



## UWS Academic Portal

### What software reuse benefits have been transferred to the industry?

Barros-Justo, Jose L.; Pincirolì, Fernando; Matalonga, Santiago; Martínez-Araujo, Nelson

*Published in:*  
Information and Software Technology

*DOI:*  
[10.1016/j.infsof.2018.06.003](https://doi.org/10.1016/j.infsof.2018.06.003)

Published: 30/11/2018

*Document Version*  
Peer reviewed version

[Link to publication on the UWS Academic Portal](#)

*Citation for published version (APA):*  
Barros-Justo, J. L., Pincirolì, F., Matalonga, S., & Martínez-Araujo, N. (2018). What software reuse benefits have been transferred to the industry? a systematic mapping study. *Information and Software Technology*, 103, 1-21. <https://doi.org/10.1016/j.infsof.2018.06.003>

#### General rights

Copyright and moral rights for the publications made accessible in the UWS Academic Portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

#### Take down policy

If you believe that this document breaches copyright please contact [pure@uws.ac.uk](mailto:pure@uws.ac.uk) providing details, and we will remove access to the work immediately and investigate your claim.

# What software reuse benefits have been transferred to the industry? A systematic mapping study.

José L. Barros-Justo<sup>a,1</sup>, Fernando Pincirolí<sup>b</sup>, Santiago Matalonga<sup>c,d</sup>, Nelson Martínez-Araujo<sup>a</sup>

<sup>a</sup> School of Informatics (E.S.E.I.), University of Vigo, (32004) Ourense, Spain

<sup>b</sup> Faculty of Informatics and Design, Champagnat University, (5501) Godoy Cruz, Argentina

<sup>c</sup> Faculty of Engineering, Universidad ORT Uruguay, (11100) Montevideo, Uruguay.

<sup>d</sup> School of Engineering and Computing, University of the West of Scotland, Paisley, United Kingdom.

## ABSTRACT

**Context:** The term software reuse was first used in 1968 at the NATO conference. Since then, work in the scientific literature have stated that the application of software reuse offers benefits such as increase in quality and productivity. Nonetheless, in spite of many publications reporting software reuse experiences, evidence that such benefits having reached industrial settings is scarce.

**Objective:** To identify and classify the benefits transferred to real-world settings by the application of software reuse strategies.

**Method:** We conducted a systematic mapping study (SMS). Our search strategies retrieved a set of 2,413 papers out of which 49 were selected as primary studies. We defined five facets to classify these studies: a) the type of benefit, b) the reuse process, c) the industry's domain, d) the type of reuse and e) the type of research reported.

**Results:** *Quality increase* (28 papers) and *Productivity increase* (25 papers) were the two most mentioned benefits. Component-Based Development (CBD) was the most reported reuse strategy (41%), followed by Software Product Lines (SPL, 30%). The selected papers mentioned fourteen industrial domains, of which four stand out: aerospace and defense, telecommunications, electronics and IT services. The application of systematic reuse was reported in 78% of the papers. Regarding the research type, 50% use evaluation research as the investigation method. Finally, 13 papers (27%) reported validity threats for the research method applied.

**Conclusions:** The literature analyzed presents a lack of empirical data, making it difficult to evaluate the effective transfer of benefits to the industry. This work did not find any relationship between the reported benefits and the reuse strategy applied by the industry or the industry domain. Although the most reported research method was industrial case studies (25 works), half of these works (12) did not report threats to validity.

**Keywords:** systematic mapping study, systematic review, software reuse, software reuse processes, software reuse benefits, real-world setting, industry, evidence-based software engineering.

## 1. Introduction

The reuse of software is a way to increase the quality of the final product [1]. If previously tested software pieces are reused in a new project, they are more likely to be error-free than newly developed ones, because of repeated use and test. This reduces the overall failure rate, hence, increasing the quality of the software artifact [2]. The rationale here is quite simple: more uses enable the discovery and removal of more bugs.

---

<sup>1</sup> Corresponding author. Tel.: +34 988 387 029

Email addresses: [jbarros@uvigo.es](mailto:jbarros@uvigo.es) (José L. Barros-Justo), [pincirolifernando@uch.edu.ar](mailto:pincirolifernando@uch.edu.ar) (Fernando Pincirolí), [santiago.matalonga@uws.ac.uk](mailto:santiago.matalonga@uws.ac.uk) (Santiago Matalonga), [nmaraujo@esei.uvigo.es](mailto:nmaraujo@esei.uvigo.es) (Nelson Martínez-Araujo).

Software reuse has received almost as many definitions as the number of authors who have written about it [3–6]. For the purposes of this paper, we will use the definition offered by the IEEE Standard for Information Technology-System and Software Life Cycle Processes-Reuse Processes[7], which is the current standard at the time of this study:

*“Software reuse entails capitalizing on existing software and systems to create new products.”*

In the definition above, the word “capitalizing” implies a reaping of potential benefits from software reuse; some key activities should be included in the Software Development Life Cycle (SDLC) to produce new systems. By considering these activities, the standard refers to *systematic reuse* and defines it as “the practice of reuse according to a consistent, repeatable process”.

Furthermore, we can split software reuse into two main processes: *reuse-with* and *reuse-for*; each containing a different set of techniques that often can (and do) overlap [8]. The main processes associated with *reuse-for* are Domain Engineering (DE) and Software Product Lines (SPL), while *reuse-with* includes Component-Based Development (CBD), Model-Based Development (MBD) and Commercial-Of-The-Shelf development (COTS).

There is a large research community working on improving software engineering through software reuse, however without good practices for ensuring the transfer into industry, the value of such research is low. Nonetheless, most of the literature about software reuse and their benefits do not report industrial applications [9–12]; many reported experiences were developed in academic environments, as pilot experiments, toy projects or theoretical proposals without empirical validation.

This work presents a systematic mapping study (SMS) aimed at identifying and classifying the benefits that software reuse has delivered to the industry. The motivations for this work are:

- To examine and characterize the extent and nature of research activity by creating a map depicting existing research of software reuse processes, but limited to those applied in the industry,
- To identify the benefits of software reuse transferred to industry,
- To uncover potential relationships between software reuse processes, transferred benefits and industry’s domains,
- To summarize and disseminate research findings to academic researchers and industry practitioners in order to better focus the activity of software reuse for maximum benefits,
- To identify research gaps in the existing literature and therefore, direct future research.

The application of the SMS resulted in a selection of 49 research articles out of a total of 2,413, which had been identified by the different search strategies. Our results show that quality and productivity increase were the two most mentioned benefits. The two most reported strategies were component-based development and software product line. Among the industrial domains having previously reported benefits and strategies, aerospace and defense, telecommunications, electronics and IT services, are the most referenced ones. The application of systematic reuse (as opposed to opportunistic reuse) was reported in 78% of the selected 49 papers.

This work can be useful to practitioners in the field of software development, as well as academic researchers. The main contribution of this work is the clear identification and, later classification, of the diverse benefits, that software reuse has contributed in the context of real-world projects (industry). Practitioners can use the results of this study as a partial guide, to evaluate if there is empirical evidence of the benefits of reuse in their organizations, and use that assessment to help the decision making process.

The rest of the study is structured as follows: section 2 presents the related work and briefly analyze the conclusions of other systematic reviews; section 3 depicts the research method and details the protocol being followed to guide the main activities (search, selection and data extraction); section 4 offers the results and also reports on the threats to validity, while section 5 analyses and discusses the findings. Finally, section 6 presents the conclusions.

## 2. Related work

As recommended by Petersen et al. [13], one of the first steps before conducting a mapping study is to identify previous systematic literature reviews or systematic mappings that are relevant to our work.

We conducted search queries in IEEE Xplore (18 results), SCOPUS (63 results) and WoS (19 results) and obtained 71 unique works (29 duplicates), from which we selected six articles that fulfilled the following selection criteria:

- Secondary or tertiary studies (systematic literature review, or systematic mapping study, or review of review),
- Published in peer-reviewed venues, and
- Focused on software reuse benefits.

Using the previous selected six works as an initial set, we performed backwards and forward snowballing [14], which add four more works to our list of articles. The following paragraphs describe the final set of these 10 selected works.

Mohagheghi's work[6] was the first attempt to review and organize the evidence about the effects of software reuse in industrial contexts. They analyzed eleven papers from journals and conferences between 1994 and 2005 to extract data about quality, productivity and economic benefits of software reuse. A significant relation between software reuse and lower problem density was found in five studies and less effort spent in correcting errors in three studies. They also found evidence of significant gains in apparent productivity (calculated by dividing the total size of software to the total effort spent) in three studies. They conclude that: a) evidence from industry is sparse; b) researchers do not use comparable metrics; c) major industry challenges include: evaluating reuse of COTS and Open Source Software (OSS) components, integrate reuse activities in software processes, evaluate Return-On-Investment (ROI) and improve data collection. Our findings confirm the mentioned benefit of lower defects density. However, we collected information on a wider range of reuse benefits. Our study included ten of the eleven studies analyzed in [6]. Compared with the other related works analyzed in this section, Mohagheghi's work is the most similar to our study.

In 2008 Mohagheghi and Dehlen [15], published a review focused on the application of MDE (Model-Driven Engineering, a synonym of MBD) in Industry. Their work covers a period from 2000 to 2007. They selected 25 empirical studies, from which they extracted data related to industry motivations for investing in MDE and the domains where the technique has been applied. Their conclusions were as follows: a) maturity of third-party tool environments is perceived as unsatisfactory; b) experiences report increases in quality; c) gains and losses in productivity were reported; d) most of the experiences were from small-scale studies, so studies on larger projects were necessary to strengthen the evidence and, finally; e) there is too little evidence on advantages of applying MDE. Our findings confirm the mentioned increases in quality and the prevalence of small-scale studies in the reports. However, our study gathered information on a broader set of reuse practices, not only MBD.

The work from Montagud [16] presents a systematic literature review to identify measures for quality attributes of SPL between 1996 and 2010. Their results show that there is little experimental evidence for the proposed measures. Out of the 35 selected papers, only 25% of the identified measurements proposal presents empirical evaluations. Another interesting result is that a vast majority of the proposed measurements (92%) deal with the "maintainability" quality attribute. The Requirements Life Cycle process is not studied by any of their selected primary works. Consequently, most measures were applied during the Design phase of the SDLC. Unfortunately, there is no guidance for practitioners to select empirically validated measures for a specific Life Cycle process of concern. Our study did not focus on measures, nor was it restricted to quality attributes or SPL. We were interested in gaining an overview on the general practice of reuse (all processes) and the benefits (all categories) transferred to the organization. However, our results, related to the quality benefit, are similar to theirs.

Kahtan [17] published a review on CBSD (Component-Based Software Development, a synonym of CBD), specifically on the challenges of security features in this type of development. They examined 26 articles on CBSD models

published between 1995 and 2012. Their focus was set on the stages of evaluation and selection of components, design and architecture and, implementation and testing, always based on security features. Their main conclusion is that existing CBSD models seem to neglect the security features and that developers need new CBSD's lifecycle process, which considers embedded security features.

Myllärniemi [18] deals with variability in SPL, specifically which quality attributes should be varied. As products in a SPL usually differ by their functionality and not their quality attributes. They conducted a Systematic Literature Review (SLR), using a method adapted from guidelines in (Kitchenham, 2004). They did not use automatic search in databases but selected the primary studies by reading through all content in all full studies published in SPL conferences. They reviewed 221 papers spanning a period of ten years from 2000 to 2010; and selected a final set of 29 primary works. Regarding their research protocol, a shortcoming is that they did not assess the quality of the primary studies and did not exclude primary studies in which the main focus was not quality attribute variability, although that was precisely their topic of interest. The quality attributes found were memory consumption, performance, security, and usability. They conclude that different approaches of feature modeling suit specific quality attributes and, that more empirical evidence on industrial quality is needed. Perhaps because the work was published in a conference (SPLC'12) and the limited number of pages (just five) the authors do not provide enough information to allow the repetition of their research. Our results cover most of the reported conclusions and confirms that the need for more industrial evidence is still a big issue.

Soares [19] presents another SLR on SPL, but they build upon the results of [16], in that their focus is on emerging Non-Functional Properties (NFP) during runtime. They selected 35 papers that fulfill their inclusion criteria and identified 52 NFPs. It is interesting to note that while their keywords are similar to the ones applied by [16], their selection of sources were different. This difference is probably due to their focus on NFP. However, Soares et al., did not define their terms of interest. Hence, the reproducibility of this research is also a weak spot. As in [16], less than half of the selected technical literature presents empirical evidence (42%, as reported by the authors). As a contribution to the body of knowledge, the authors group the emerging NFP in three categories: feature selection, estimation approaches, and prediction approaches. The design of our study, although similar, is based on a different objective, which gives rise to different research questions. Our results do not focus on NFP, so we cannot compare the outcomes.

The work from Flemström [20] deals with a very specific research area: vertical test reuse for embedded systems. They defined vertical reuse as the reuse of artifacts over different integration levels during the engineering process. They conducted a SMS to gather information about the state of the art in vertical test reuse for the embedded systems domain. Eleven papers were analyzed, from 2005 to 2015. They found four main motivations for vertical test reuse: avoiding rework for similar test cases, increasing quality of test cases, avoiding costly creation of new environments and reducing test complexity. Abstract test cases and test models were reported as the most frequent reused items. Three reuse approach classes were identified: adapter-based, formal methods and clone mining-based.

Varnell et al. [21] performed a survey of software reuse, comparing reuse outcomes for embedded systems versus non-embedded systems using different development approaches. They selected 84 candidate studies and classified them by study type, system type (embedded and non-embedded) and development approach. After applying exclusion criteria, they finally analyzed 43 works. The period considered was from 1992 to 2013. Like many other authors, they conclude that there are few reported studies from industry. Papers with enough evidence (quantitative) are scarce and some authors even question whether the benefits exist or not [12,22]. Existing studies lack solid metrics to assess the many opinions about software reuse and its benefits. The authors also found that results from reuse of non-embedded systems were not necessarily extendable to embedded systems. The implementation of the research method by Varnell et al. has several weaknesses, including amongst them the absence of rules for inclusion/exclusion and the lack of a report (or at least the references) with the selected works. Our study identifies similar issues and partially confirms the differences in reuse outcomes from embedded and non-embedded systems.

Goulão's work [23] reported a tertiary study about quality in model-driven engineering. They identified 22 systematic literature reviews and mapping studies that report quality attributes in the context of MDE, from 2008 to 2015. The data extracted include information from the publication space (authors, affiliations, countries, year and so) and the research space [13], such as quality models and quality attributes addressed in secondary studies. The quality model and quality attributes were categorized by using the standard ISO/IEC 25010-2011[24]. All of the 22 selected studies were targeted to researchers and only five were addressed to practitioners. Maintainability was the most frequently reported quality attribute (18 studies), followed by reliability, efficiency, and usability (5 studies). The authors concluded that "the coverage of quality attributes is wide, but with a predominance of product quality attributes over quality in use attributes" [23]. Our findings confirm these results, however, the scope and the study design differs. Our study select only primary papers, while the work of Goulão use secondary studies as its data source.

Finally, Irshad et al. [25] conducted a SLR and an industrial evaluation of cost avoidance through reuse. Cost avoidance captures benefits that are different from cost savings (direct savings), such as the costs avoided because of defect prevention (quality aspects of the product). The work has two main objectives: a) to assess the status of capturing cost avoidance through reuse in the academia and, b) to propose improvements, integrate these into an instrument and, evaluate the instrument in the industry. The SLR was conducted to achieve the first objective. They found and analyzed only three studies and none of these three studies was validated in industry. The authors did not report the period of their study, and only an automated search strategy was conducted. From the SLR the authors concluded that: a) the number of lines of source code is not a relevant metric for many reusable artifacts, b) all models calculate reuse cost avoidance by the formula Cost without reuse - Cost with reuse. However, the metrics differ between models, c) no guidelines on what metrics to collect for calculating cost avoidance were found and, d) the scientific literature contains evaluation experiences of only two models (two studies, one primary study was not evaluated in industry).

Table 1 Summary or Related Works

Study	Reuse processes	Type of benefit	Up to (Year)	Context
[6]	Open list (all processes)	Quality, Productivity and Economic benefits	2005	Industry
[15]	MBD	Quality and Productivity	2007	Industry
[16]	SPL	Quality	2010	Not considered
[17]	CBD	Not reported	2012	Not considered
[18]	SPL	Quality (variability)	2010	Not considered
[19]	SPL	Non-Functional Properties	2013	Not considered
[20]	CBD	Quality and Productivity	2015	Not considered
[21]	Open list (all processes)	Quality and Productivity	2013	Not considered
[23]	MBD	Quality	2015	Not considered
[25]	Not reported	cost avoidance (measurement)	Not reported	Not considered

Table 1 presents a summary of the findings from this literature evaluation. Our work differs from the works analyzed in this section in the following aspects:

- it focuses on industrial settings, where a lack of data is reported,
- all the reported benefits and reuse processes were included (open list), and
- the period considered in our work is up to 31st of December 2016.

### 3. Research method

We followed the guidelines in [13] to plan and conduct this SMS. Figure 1 depicts the overall process. For the planning phase, a detailed description of the research protocol was produced and can be accessed online[26]. The next sections present the general goal and research questions and details the steps of the conducting phase.

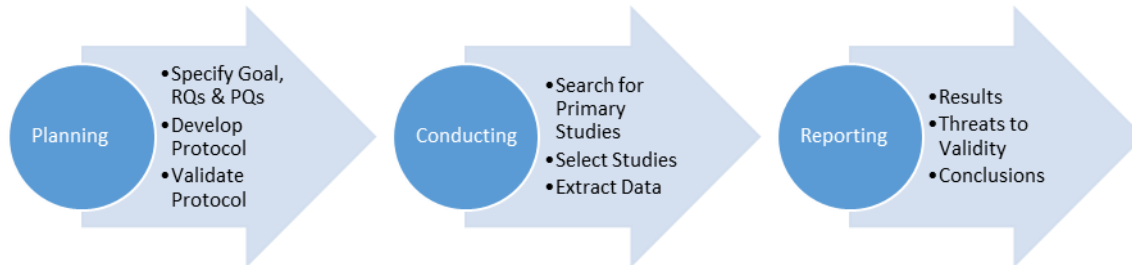


Figure 1 Diagram of the research phases and steps in this study.

#### 3.1. Specify Goal and Research Questions

The goal of this mapping study was established as:

*Identify and classify the benefits that software reuse has transferred to the industry*

From this goal, we derived a set of Research Questions (RQ) (explained in Table 2) and Publication Questions (PQ).

##### Research space questions

- RQ1: What benefits have been transferred to the industry?
- RQ2: Which reuse processes are used?
- RQ3: Which are the industry's domains?
- RQ4: Which type of reuse is reported?
- RQ5: Which research types were applied?
- RQ6: Were threats to validity reported?

##### Publication space questions

- PQ1: Which are the Top venues?
- PQ2: How many publications per year were identified?
- PQ3: Who are the most active researchers?
- PQ4: What is the affiliation of these Researchers?
- PQ5: Which are the most active countries?

Table 2 Description of the Research Questions

Research Question	Description
RQ1: What benefits have been transferred to the industry?	Name of the benefit (or benefits), as it appears in the original source (verbatim).
RQ2: Which reuse processes are used?	Name of the reuse processes reported (verbatim)
RQ3: Industry's Domain	The global industry classification standard (GICS) [27]

<i>RQ4: Which type of reuse is reported?</i>	Systematic or Planned Reuse Versus Ad Hoc or Opportunistic Reuse.
<i>RQ5: Which were the research types?</i>	The types of research proposed by Wieringa et al., [28]. As our study focuses on results in industrial contexts, the types Validation research, Philosophical and Opinion papers will be excluded, because they do not fulfill the condition “Used in practice”. See Table 3.
<i>RQ6: Were threats to validity reported?</i>	Is there a section reporting threats to validity? Y/N? (if Yes, then which threats were reported?)

Table 3 shows the set of decision rules, columns R1 to R6, were used to help to decide whether a primary paper reports on one of the research types proposed by Wieringa et al. [28]. For example rule R1 means that to classify a source as an “Evaluation research” (Decision – bottom half of Table 3), the source should:

- satisfy the first and third conditions,
- do not satisfy the fifth condition,
- should be considered irrelevant for the second, fourth and sixth conditions.

Table 3 Research type classification (T = True, F = False, \*= irrelevant, R1-R6 refer to decision rules). Adapted from [13]

	R1	R2	R3	R4	R5	R6
<b>Conditions</b>						
Used in practice	T	F	F	*	F	T
Novel solution	*	*	F	T	F	F
Empirical evaluation	T	T	F	F	F	F
Conceptual framework	*	*	T	*	F	*
Author’s opinion	F	F	F	F	T	F
Author’s experience	*	*	F	*	F	T
<b>Decisions</b>						
Types of research[28]	Evaluation research	✓				
	Validation research		✓			
	Philosophical papers			✓		
	Solution proposal				✓	
	Opinion papers					✓
	Experience papers					✓

Table 4 presents a brief description for each publication space question (PQ).

Table 4 Description of the Publication Questions

<i>PQ1: Top venues</i>	Name of the original source: Conference, Journal or Workshop
<i>PQ2: Publications per year</i>	Evolution, publications per year (no matter the venue)
<i>PQ3: Active researchers</i>	All authors (of papers in our set of selected works), ordered by number of papers
<i>PQ4: Researcher’s affiliation</i>	All authors, Academic or Industry or Both (at the time of the publication)
<i>PQ5: Active countries</i>	Based on the author’s affiliation



The following section describes the search protocol and provides details of how the search string was incrementally built. The first author, as a domain expert, was responsible for verifying the validity and appropriateness of the terms in the search string.

### 3.2. Search for primary studies

The search consists of three different strategies: *Automatic search* run in publishers and indexers databases (section 3.2.1), *backward and forward snowballing* from the studies selected for our related work section (section 3.2.2) and, a *manual search* of published papers from the main conferences related to software reuse (section 3.2.3). Figure 2 summarizes the search process and results obtained through each of these activities. The next three subsections detail each of the search strategies.

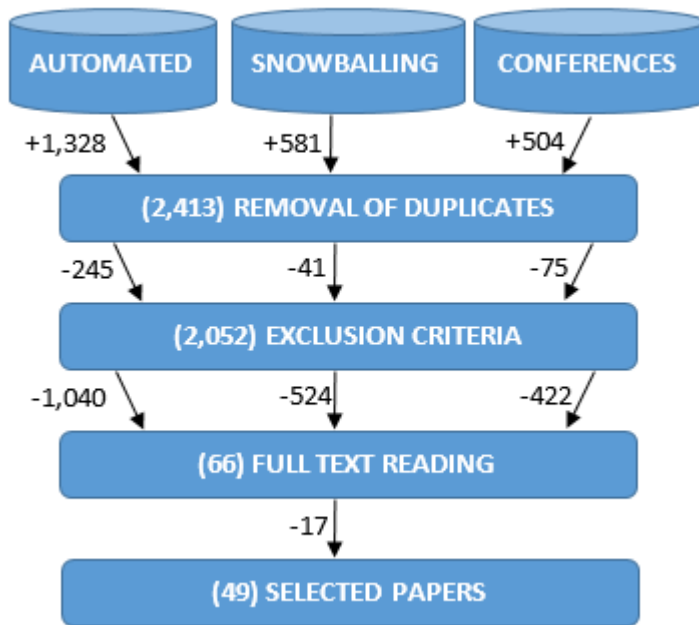


Figure 2 Results from the search strategy

#### 3.2.1. Automatic search

To conduct the automatic search, a search string was built. Special care was taken in this step, since it has been reported that systematic reviews fail because of an inadequate search string construction[29–31].

To create the search string, we first derived the major search terms from our main goal and the set of RQs and PQs. After that, we conducted a pilot search in SCOPUS, using these major terms to identify other relevant keywords, synonyms and alternative spellings. The key terms came from two different sources, authors and editors. Finally, we applied the PICOC method (Population, Intervention, Comparison, Outcome and Context), as suggested by Kitchenham and Charters [32] and connect the resulting terms using Boolean operators.

The key terms derived from the goal, RQs and PQs were:

- Goal: *software reuse*.
- RQ1: *quality, productivity*.
- RQ2: *domain engineering, software product line, component-based, model-based, model-driven and commercial-off-the-shelf*.
- RQ3: *industry, firm, business, and company*.
- PQ1: *conference, journal, and workshop*.

The pilot search added two derived terms: *Reusability and reusable*.

We applied some restrictions (Limits) to the searches to narrow the results. These are a filter by *type of venue* and *subject area*.

The combination of PICOC and Limits produces the following set of key terms:

- Population: “software reuse”, “domain engineering”, “software product line”, “component-based”, “model-based”, “model-driven”, commercial-off-the-shelf.
- Outcomes: quality, productivity, reusability.
- Context: industry, firm, business, company
- Limits: journal article, conference paper, subject area: computer science

The final search string was:

*(“software reuse” OR “domain engineering” OR “software product line” OR “component-based” OR “model-based” OR “model-driven” OR commercial-off-the-shelf) AND (quality OR productivity OR reusability) AND (industry OR company) AND (journal article OR conference paper) AND subject area (software engineering)*

Finally, we tailored the search string to each of the searched resources. Table 5 shows the final syntaxes of the search string for each database, given their particular query language and restrictions.

Table 5 Search string syntaxes

Database	Search string
IEEE Xplore	((("Document Title":reuse OR "Document Title":domain engineering" OR "Document Title":software product line" OR "Document Title":component-based" OR "Document Title":model-based" OR "Document Title":model-driven" OR "Document Title":commercial-off-the-shelf) AND (Search_Index_Terms:reusability OR "Index Terms":quality OR "Index Terms":productivity) AND (Search_Index_Terms:industr* OR "Index Terms":compan*))
SCOPUS	TITLE(reuse OR "domain engineering" OR "software product line" OR "component-based" OR "model-based" OR "model-driven" OR commercial-off-the-shelf) AND TITLE-ABS-KEY(reusability OR quality OR productivity) AND TITLE-ABS-KEY(industr* OR compan*) AND (DOCTYPE(ar) OR DOCTYPE(cp) ) AND ( LIMIT-TO(SUBJAREA,"COMP " ) )
Web of Science	#1: TI=(reuse OR "domain engineering" OR "software product line" OR "component-based" OR "model-based" OR "model-driven" OR commercial-off-the-shelf) #2: TS=(reusability OR quality OR productivity) #3: TS=(industr* OR compan*) #4: #1 AND #2 AND #3 #5: #1 AND #2 AND #3 Refined by WEB OF SCIENCE CATEGORIES:(COMPUTER SCIENCE SOFTWARE ENGINEERING)

The execution of the above search strings in the databases retrieved 1,328 papers.

### 3.2.2. Backward and forward snowballing search

Snowballing refers to the process of using the reference list of a paper (backward snowballing) or the citations to the paper (forward snowballing) to identify additional papers [14]. We conducted a snowballing search (backward and forward) using the papers in our section of related work (10 secondary works) as the initial set (seeds), following the guidelines proposed in [14].

The motivation for the selection of the related works as the initial set was that these reviews offer a list of papers (their selected primary papers) that are of potential interest for this study. The backward strategy retrieved 453 works, while another 128 came from the forward process. We removed thirty duplicates. The execution of the snowballing search strategy retrieved 581 papers. Table 6 presents the summary of this search strategy.

Table 6 Snowballing search

Paper_ID	Related work	References	Citations	Duplicates
AS1	[17]	68	4	1
AS2	[16]	55	12	---
AS3	[18]	37	1	---
AS4	[25]	52	0	1
AS5	[6]	57	59	7

AS6	[20]	28	0	4
SB1	[19]	16	1	2
SB2	[23]	56	1	4
SB3	[21]	53	0	3
SB4	[15]	31	50	8
	Total works	453	128	30

### 3.2.3. Manual search on selected international Conferences

A manual search was conducted on the three major international conferences related to software reuse. We ran the search in SCOPUS using the search field CONFNAME. To confirm the results of this query, we also conducted a complementary search in Google Scholar, retrieve the table of contents of every conference edition and manually count the number of papers. The execution of this strategy retrieved 504 papers. Table 7 summarizes the results.

Table 7 Conference's search

Conference (and time frame)	Retrieved papers
ICCBSS (International Conference on Commercial-off-the-Shelf (COTS)-Based Software Systems, 2002-2007; International Conference on Composition-Based Software Systems, 2008)	135
ICSR (International Conference on Software Reuse, 1996-2015)	276
SPLC (Software Product Line Conference, 2004-2014)	93
Total works retrieved	504

### 3.3. Study selection and Exclusion criteria

We first eliminated duplicates in the set of retrieved papers. This preliminary task ensures that we do not analyze the same paper more than once. After that, all papers were collected into a single spreadsheet from every search strategy and a two-step process was carried out:

Step 1: the results were screened on title, keywords, and venue (Journal or Conference) only, to exclude papers from any of the following categories:

- Not in English,
- Not peer-reviewed publications,
- Not a primary study (secondary/tertiary studies were evaluated in our related work section),
- No major search terms (from PICOC) appears in Keywords (from author or editors)
- Books, editorials, tutorials, panels, poster sessions, prefaces, opinions, letters, slides and any work that can be considered as grey literature,

Step 2: Papers from Step 1 were evaluated on abstracts and, if necessary on a full-text reading, to exclude the studies that were:

- Not within the scope: these are irrelevant papers that were retrieved due to poor execution of the search or because of human errors in manual searches. We found this exclusion criterion the most effective one. The “not within scope” criterion was subdivided into three related criteria:
  - Not about software reuse (or any software reuse processes) [Not in Population]
  - Not about any software reuse benefit [Not in Outcome]
  - Not an industrial setting [Not in Context]
- Limits: papers not from the domain of IT/CS/SE/IS (Information Technology/Computer Science/Software Engineering/Information Systems)
- Others: We also excluded PhD or Master Theses, under the assumption that relevant publications, resulting from the research covered by the theses, were available and included in the set of retrieved papers.

Finally, for studies having conference and journal versions, we select the most recent (normally the Journal version);

if these publications were equal, then we selected the most detailed one.

The application of the exclusion criteria was done by:

- One reviewer, to conduct the removal of duplicates (first author)
- A pair of reviewers, to conduct Step 1 (first and second author)
- Two pairs of reviewers, to conduct Step 2: one pair (second and third author) analyzed the first half of the selected papers from Step 1 and the second pair (first and fourth author) the other half. On a second round, the pairs interchange their halves.

To deal with disagreements, we applied the inclusive criteria proposed in [13]. We excluded a paper only when both reviewers agreed (category “F”) or considered the paper as borderline (category “E”) (see Table 8).

Table 8 Dealing with disagreements

		Pair of reviewers X		
		Include	Uncertain	Exclude
Pair of reviewers Y	Include	A	B	D
	Uncertain	B	C	E
	Exclude	D	E	F

Table 9 show the final result of the selection process, a set of 49 selected papers.

Table 9 The Selection results for each step in the process.

Source	#works	Selected Works (PaperID)
IEEE Xplore	5	S1-S5
SCOPUS	20	S6-S25
WoS	4	S26-S29
Conferences	6	S30-S35
Snowballing	14	S36-S49

### 3.4. Data extraction

We developed a spreadsheet Data Extraction Form (DEF) to gather all the data extracted from the selected primary works. A template was used to organize the items of interest. Reviewers who performed the data extraction process were constrained to choose only one option from the list of possible values to answer every column (RQs) and, should add a comment containing a rationale for their selection, including the exact location (page) in the original work of the data extracted. Allowed values for reuse benefits (RQ1) originally came from the IEEE standard [7], but during the process of data extraction, we also considered synonyms or other significant terms not included in it. The data extraction form is presented in Appendix 3.

In addition to this, we devised a protocol to reduce individual bias during the data extraction process. This protocol contained the following steps:

1. The set of selected primary works should be divided into two halves (H1 and H2),
2. Two reviewers (r1 and r2), independently, will extract data from H1,
3. A reviewer (r0) will compare outputs from reviewers, integrate results and resolve possible conflicts,
4. Step 2 is repeated for reviewers r2 and r3 working in H2. Then step 3 will be executed again.

## 4. Results

The results from the data extraction process is reported in this section. The 49 selected papers resulting from the selection process (section 3.3) are [11,33–80], they are presented (grouped) in Annex I.

Only the relevant data for the research and publication questions are shown. The analysis of these results is

presented in section 5.Discussion.

## 4.1 Research Space questions

### 4.1.1. RQ1: Which benefits have been reported?

According to the outcome from the data extraction process, there are 128 mentions of 26 different benefits in the selected studies. Of these benefits, *Increase Quality* and *Increase Productivity* are the two most mentioned benefits (Table 10). Another 14 benefits are mentioned only once, and from these, four are considered disbenefits (in italics, at the bottom of Table 10).

Table 10 Papers mentioning benefits

RQ1 (Benefits)	Paper ID
Increase Quality	S1, S3, S5, S7, S8, S9,S10, S14, S15, S16, S18, S19, S21, S25, S26, S27, S28, S29, S35, S36, S37, S39, S42, S43, S46, S47, S48,S49
Increase Productivity	S1, S3, S4, S5, S9, S10, S13, S14, S15, S19, S21, S22, S23, S25, S27, S29, S33, S36, S37, S39, S43, S44, S46, S48, S49
Reduce Development Cost	S1, S2, S3, S5, S8, S14, S16, S19, S21, S24, S28, S29, S34, S35, S39, S44, S45
Reduce Development Time	S1, S4, S6, S8, S18, S19, S20, S23, S24, S29, S30, S32, S34, S35, S39
Lower Defects Rate	S9, S10, S11, S26, S32, S40, S41
Shorter Time to market	S14, S19, S23, S25, S28
Reduce Effort	S11, S17, S18, S40
Increase Efficiency	S16, S38, S44
Increase Maintainability	S8, S18, S38
Increase Portability	S1, S22
Increase Standardization	S8, S22
Reduce Test Time	S11, S25
Automation	S47
Knowledge sharing	S8
Platform independence	S22
Reliability	S31
Code size	S12
Upgradeability	S38
Evolution time	S18
Customer satisfaction	S42
Stable components	S41
Product safety	S45
<i>Hidden cost of COTS/GOTS</i>	S30
<i>No reduction of development Cost</i>	S34
<i>No reduction of Time to Market</i>	S34
<i>Performance Decay</i>	S9

The work from Incorvaia and Davis [S1] [33], in 1990, is one of the most comprehensive report of case studies in software reuse. They analyzed six reuse projects in five different organizations (Digital Equipment Corporation, BTG Inc., U.S. Army, Raytheon Company and Hartford Insurance). Their goal was to focus on the identification of attributes that make reuse successful; to do that, they tested 18 hypothesis, the 18<sup>th</sup> was that “Reuse improves productivity and quality”. Only Raytheon and Hartford reported increases in productivity. The other three organizations did not collect data; the authors said that, perhaps the managers and developers believed the 18<sup>th</sup> hypothesis to be self-evident. “Studies have not been conducted on whether hammers work better than rocks. We all know hammers work better” [S1].

Conradi, Mohaghegi, Gupta and Slyngstad, conducted an empirical study in 2006, with the participation of the IT department of a large oil and gas company in Norway (Statoil ASA) [S8] [40]. They used a survey and semi-structured interviews to gather information about the developer’s views on software reuse. All 16 participants considered reuse as beneficial, as well as incurring lower cost, shorter development time and higher quality of the reusable components as the main benefits. An interesting discovery of their study was that “quality attribute specifications were trusted for the applications using reusable components in new development, but not for the reusable components themselves.” [S8].

Sellier et al., presented an interesting report in [S18] [49] that mixes academic researchers, researchers from the

European Software Institute and workers from MSIGrupo<sup>2</sup>. They described the adoption process of the Software Product Line Engineering (SPLE) approach for the development of an embedded software system for metal processing lines. The adoption of SPLE reduced the development time and effort, as well as its product maintenance effort. Another benefit reported by the authors was that derived from a better product understanding, which in turn reduced the risk to lose knowledge, but they also highlighted that “SPLE had to be applied progressively and iteratively using a company sub-domain as a part of the transfer of technology.” [S18].

Some disbenefits were also identified. An article reporting the integration of COTS/GOTS within NASA’s Hubble Space Telescope command and control system [S30] identified the many hidden costs and risk of COTS/GOTS, specifically the need for a careful selection process, consulting fees and training. Kolb et al. [S34] also report disbenefits when dealing with product line development. The authors assessed the consistency and quality of a product line for climate and flue gas measurement devices. They did not find reductions in development time or maintenance costs. However, the experience was considered successful, because of the satisfaction of the involved developers and the increment in their ability to develop products that were more complex.

Deniz et al., [S9] [11] conducted an empirical study within the software engineering department of a Turkey’s defense industry. They collected a large data set in three different case studies and found that, although reuse produced benefits, the performance of some systems decreased, due to the use of reusable components, which were not designed or optimized for the target system.

We also analyzed the number of times that the four most reported benefits (Quality, Productivity, Development Cost and Development Time) appeared over the years (Figure 3). The distribution of benefits over time is fairly uniform; no benefit stands out, in excess, with respect to the others.

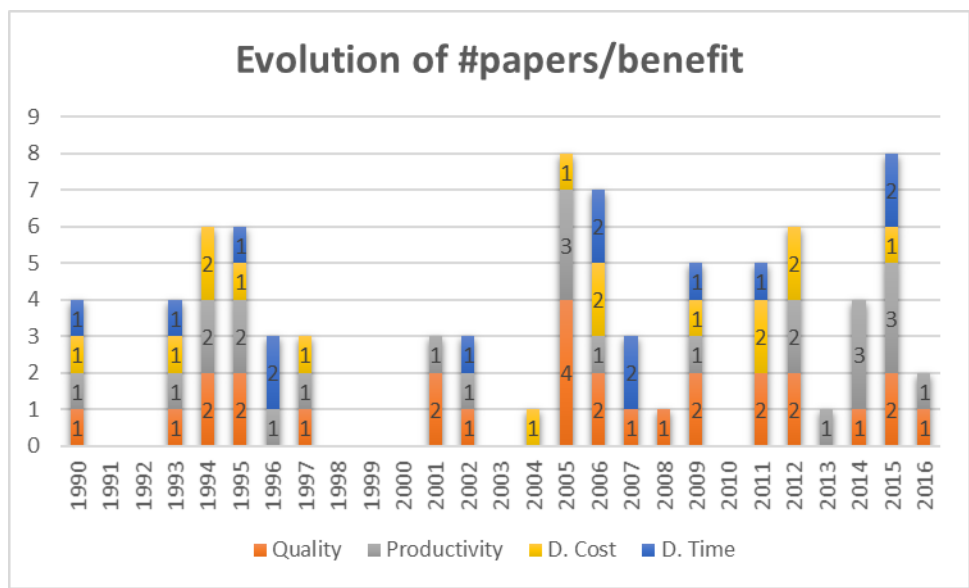


Figure 3 Mentions to benefits and their evolution in time

#### 4.1.2. RQ2: Which reuse processes are used?

Our study identified 65 mentions of reuse processes (some papers report more than one case study). CBD (47%) and SPL (27%) were the two most reported processes.

Table 11 Reuse processes

RQ2 (Processes)	Paper ID	#papers	#mentions
CBD	S1, S3-S8, S10-S12, S14, S19, S21, S24-S28, S29, S36, S37, S39-S42, S44	26	31
SPL	S2, S9, S11, S16-S18, S21, S23, S29, S33-S35, S43, S46	14	18
MBD	S11, S13, S15, S21, S22, S32, S38, S45, S47-S49	11	11

<sup>2</sup> <http://msigrupo.com/en/>

COTS/GOTS	S11, S30, S31	3	5
Not reported	S20	1	---

CBD is a reuse-based approach that involves the definition, implementation and integration of components into a new system. These components may have been developed for the purpose of reuse (reuse-for) or come from other systems in use (reuse-with). The basic principle of the CBD is the idea that similar functionalities can be useful in different situations, making the components that offer them reusable [81].

Frakes and Succi [S10] [41] presented a detailed research on the reuse of code modules in an industrial context. The authors found a direct relationship between the amount of reuse and quality, while the results for productivity were ambiguous, in line with [S9, S30] [11,61]. On the other hand, the work from Ha et al., [S3] [35], who study the relationships between reuse rate against productivity, cost, quality and time, in the context of embedded software, concludes that higher modular reuse can improve productivity, cost and quality. An example of reuse-with is presented by Mohagheghi and Conradi in [S7] [39] in the context of a large telecom product. They report a significant lower fault density and less modified code between successive releases of the reused components.

SPL is defined in the standard ISO/IEC 26550:2015 [82] by means of two life cycles, domain engineering and application engineering. A SPL “consists of a set of products and/or services sharing explicitly defined and managed common and variable features and relying on the same domain architecture to meet the common and variable needs of specific market.” [82].

Kolb et al., describe their experiences with SPL at Testo AG in [S34] [65]. The focus of their study was to assess the success, consistency and quality of Testo’s product line. Although their results did not show a quantifiable reduction of time-to-market or development or maintenance costs, the authors considered the introduction of the product line as a success. The success was associated with Testo's ability to develop more complex products, and the increase in the satisfaction of the developers involved in the project. Another interesting experience, from Hitachi, is reported in [S33] [64] by Kodama et al. The authors describe an estimation method for the integration of heterogeneous clinical instruments in the CIIMS system (Clinical Instrument Integration Management Software). The method estimates commonality with precision, and allows the integration of new instruments into the system through an SPL process. The results showed an increase of 2.5 in productivity. Finally, the work of Quilty and Cinnéide [S16] [47] reports experiences with SPL in Risk Management Software. The analysis of the data collected during 10 years showed an increase in efficiency and quality, as well as a reduction in costs.

MBD promotes the reuse of models or prototypes. The goal is to increase the level of abstraction of the development process thereby decreasing the complexity and facilitating the potential automation [S15] [46].

The MBD experiences at Motorola are reported in [S48] [79] and [S49] [80]. Baker et al., reported more than 15 years of experiences in deploying a top-down approach to MDE (Model- Driven Engineering) in. They found that through the coordinated and controlled introduction of MDE techniques, significant quality and productivity gains could be consistently achieved [S48]. On the other hand, Weigert and Weil reported that "Model-driven engineering has dramatically increased both the quality and the reliability of software developed in our organization, as well as the productivity of our software engineers." in [S49]. Other experiences include the adoption of MBD in a large financial organization [S47] [78] and in a railway-signaling manufacturer [S45] [76], a detailed study of the level of adoption and use of MBD techniques in the Italian industry [S22] [53] and, the assessment of the effects of MBD in productivity at Robert Bosch Engineering, a CMMi level 5 software company [S13] [44].

COTS/GOTS are Commercial or Government Of the Shelf products, procured for integration into software systems [S11] [42]. Two studies from NASA, one from the Hubble Space Telescope command and control system (CCS) [S30] [61], and the other from the Goddard Space Flight Center [S31] [62], report on the integration of COTS/GOTS into software systems. More than 30 COTS/GOTS products allowed the new CCS to exceed the capabilities of the original system, while reduced the lines of custom code by more than 50%. On the other hand, their data also showed the high cost of COTS/GOTS solutions including training expenses, consulting fees, and long-term maintenance expenses

[S30]. A close relationship with the seller, the previous evaluation of test versions, the modular design and the constant tests, allowed mitigating the risks and concerns associated with the use of COTS in the development of a new Standard Autonomous File Server (SAFS), in support of NASA's satellite ground network. The successful integration of COTS products into SAFS convinced NASA to accept it as a standard for file distribution. [S31].

Finally, Figure 4 shows the evolution of the quantity of papers mentioning reuse processes. The graph shows that interest in reuse has been maintained over time, and that the number of papers in recent years (since 2011), is evenly distributed between SPL (7 papers) and MBD (5 papers), with a lower proportion of CBD (3 papers) and a single paper about COTS/GOTS. It should be understood that this distribution obeys the limitations imposed by our selection process (selection criteria) and does not represent, in any case, other publications in the area of software reuse.

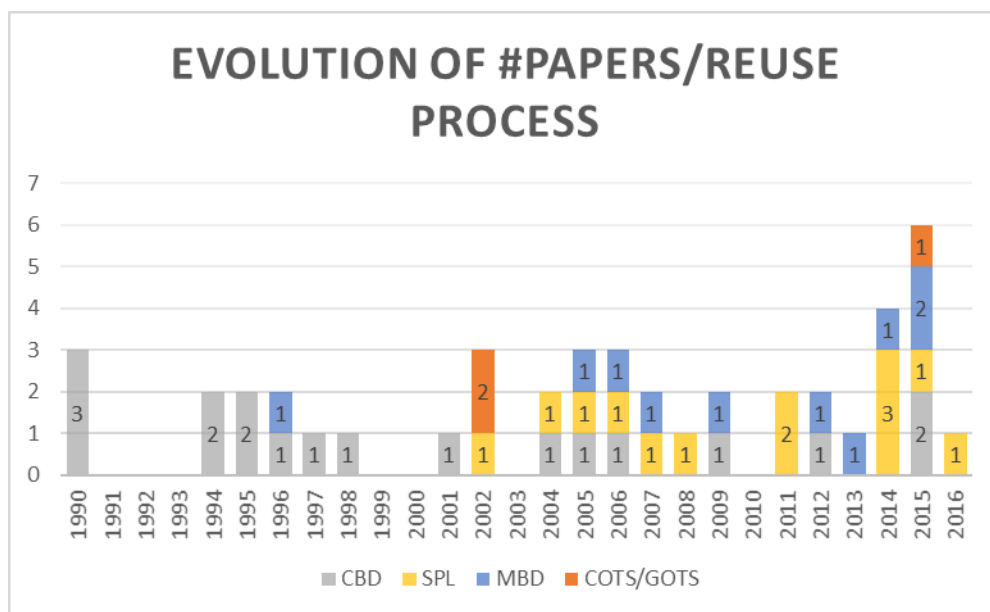


Figure 4 Mentions to processes and their evolution in time

#### 4.1.3. RQ3: Which are the industry's domains?

To classify the industries mentioned in the selected papers, we used the global industry classification standard (GICS) [27]. The GICS was developed by Morgan Stanley and Standard and Poor's. The last change to GICS includes 11 sectors, 68 industries and 157 sub-industries [27]. We started by extracting the data (verbatim) from the original work and developed an open list, which was later mapped to the GICS standard. The most cited industry's domain is aerospace and defense, followed by telecommunications, electronic equipment and IT services. As we can see in Table 12 one single paper (for example, S1) can include several mentions to different industries, and therefore to several domains, in this case we have indicated all domains reported (count every mention to every different domain).

Table 12 Industry's domain

RQ3 (Domain)	Paper ID	#papers
Aerospace & Defense	S1, S6, S9, S11, S23, S30, S31, S37, S40	9
Diversified Telecommunication Services	S5, S7, S12, S17, S41, S44, S48, S49	8
Electronic Equipment, Instruments & Components	S1, S3, S18, S25, S28, S34, S38	7
IT Services	S1, S13, S15, S16, S19, S46, S47	7
Software	S2, S35, S39, S42	4
Not reported	S21, S22, S24, S27	4
Information Technology	S14, S20, S29	3
Energy	S8, S26	2



Government contractor	S4, S10	2
Insurance	S1	1
Internet Software & Services	S36	1
Technology Hardware & Equipment	S33	1
Technology Hardware, Storage & Peripherals	S32	1
Transportation Infrastructure	S45	1
Wireless Telecommunication Services	S43	1

Table 13 offers a summary of the most representative organizations in each domain. The year of the study is shown in parentheses. Not all the papers reported the company's name, as some organizations prefer to remain anonymous ("Not reported" row in Table 12).

Table 13 Representative organizations

Domain	Organization
Aerospace & Defense	U.S. Army ISEC (1990); Raytheon Company (1990); NASA: Flight control systems (2007), Spacecraft control (2005), Satellite ground network (2002), Hubble Space Telescope command and control (2002), Goddard Space Flight Center (1997); Aselsan Electronics (2014), Israel Aerospace Industries (2015)
Diversified Telecommunication Services	Nippon Telegraph and Telephone Corporation (1995); Ericsson (2004, 2008); Wikon GmbH (2008); Philips Kommunikations (1996); Motorola (1994, 2005, 2006)
Electronic Equipment, Instruments & Components	Digital Equipment Corporation (1990); Testo AG (2006); Orbotech (2005); Freescale Semiconductor (2012, 2015); Mondragon (2007); Computime Ltd. (2012)
IT Services	IBM (2016); Matra Cap Systemes (1995); ORisk Consulting (2011); Business Solutions (2014); Robert Bosch Engineering (2014); BTG Inc. (1990); Metaphor Vision Ltd. (2007)
Software	Fujitsu (2011); Israel Software Reuse Industrial Consortium (2004)
Information Technology	Hewlett-Packard (1993, 1994, 1995)
Energy	Statoil ASA (2006); StatoilHydro ASA (2009)
Government contractor	BTG Inc. (1990); PRC Inc. (1996, 2001); Software Uno (2001)
Insurance	Hartford Insurance (1990)
Internet Software & Services	Microchip Pvt. Ltd (2015)
Technology Hardware & Equipment	Hitachi Ltd (2014)
Technology Hardware, Storage & Peripherals	Storage Technology Corp. (1996)
Transportation Infrastructure	Anonymous: Railway Signaling manufacturer (2012)
Wireless Telecommunication Services	Anonymous: network division of an italian development company (2002)

For the four most cited domains, we analyzed the evolution along the years of papers citing them. Telecommunications, an active domain from 1994 to 2008, seems to have lost interest in recent years, while IT services and aerospace and defense have maintained citations levels throughout the period analyzed.

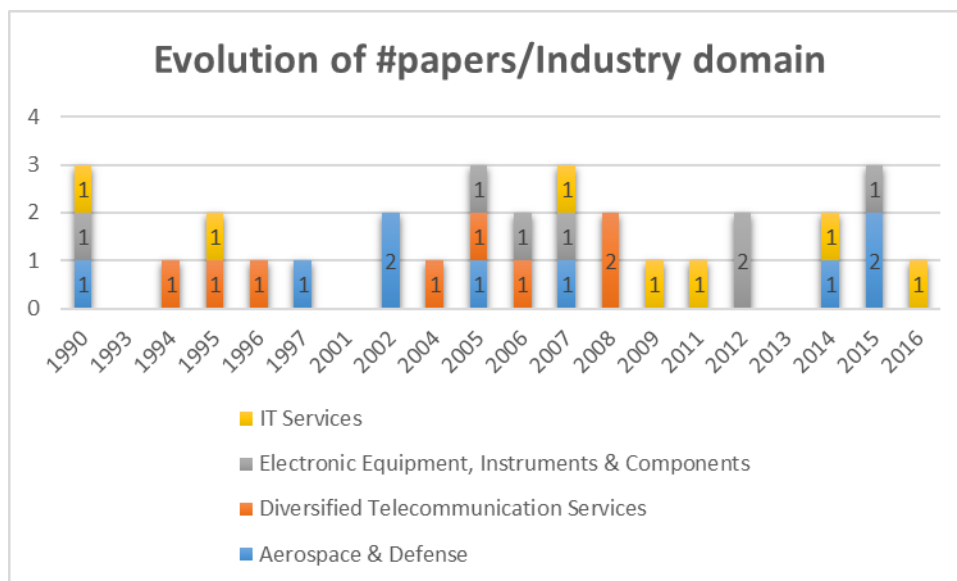


Figure 5 Mentions to Industry's domain and their evolution in time

#### 4.1.4. RQ4: Which type of reuse is reported?

Ad hoc reuse, also known as opportunistic or unplanned reuse, can be defined as "reuse performed with little or no

planning or commitments to produce, broker or consume assets” [83]. During the development of a project, the developers realize that there are artifacts that can be reused, because they provide all or part of the required functionality. The reuse of these artifacts can occur with or without transformations/adaptations of the original artifact (reuse-with).

Systematic reuse, also known as planned reuse, occurs when the organization strategically plan for reuse, and designs artifacts with the intention of being reused in future projects (reuse-for).

Systematic reuse was mentioned in 38 papers (78%) and Ad-hoc reuse in seven (14%). Four papers did not report the type of reuse applied. Some papers (multi-case studies, for example S1 and S11) reported both types of reuse.

The proportion of papers that report systematic reuse, with respect to those who confess to having used ad hoc reuse, is in line with the recommendations of the researchers. However, it is also recognized, by the research community, that the reporting/publishing of positive results is more frequent, and that, sometimes, negative results are hidden, or not reported at all.

Table 14 Type of reuse

RQ4 (Type of reuse)	Paper ID	#papers
Systematic/Planned	S1, S4, S5, S8, S11, S13-S18, S20, S21, S23, S25-S41, S43-S49	38
Ad Hoc/Opportunistic	S1, S2, S6, S10, S11, S19, S42	7
Not reported	S7, S9, S22, S24	4

#### 4.1.5. RQ5: Which research types were applied?

The most mentioned research type was Evaluation research, with 25 mentions that used case studies as research method and six using surveys (summing up 31 mentions, 62%). The final number of occurrences is 50, because the work S13 (Evaluation research) mentions the two methods. Our set of selected papers does not contain any work classified as Solution Proposal, although this type of research is very frequent in the literature. We believe that the absence of Solution Proposals is due to our exclusion criterion, which rejected works that did not deal with an application in a real-world context (industry).

Table 15 Type of research reported

RQ5 (Research type and method)	Paper ID	#papers
Evaluation research (Case Study)	S1-S7, S9, S12-S17, S23, S26, S27, S36, S37, S39-S43, S46	25
Experience report	S18-S20, S25, S28-S35, S38, S44, S45, S47-S49	18
Evaluation research (Survey)	S8, S11, S13, S21, S22, S24	6
Quasi-Experiment	S10	1

#### 4.1.6. RQ6: Threats to validity

Almost 74% of the selected papers did not report threats to validity. Remarkably, only 10 (40%) of the 25 papers that used the Case Study research method reported validity threats, while only three (S8, S11, S22) out of the six papers that utilized Survey methods reported such threats.

Table 16 Papers reporting threats to validity

RQ6	Paper ID	#papers
No	S2-S6, S10, S12-14, S16-21, S24, S25, S27-S40, S44, S45, S47-S49	36
Yes	S1, S7-S9, S11, S15, S22, S23, S26, S41-S43, S46	13

## 4.2 Answer to Publication Space questions

### 4.2.1. PQ1: Publication venues: Conferences and Journals

Our data shows that the papers are evenly distributed between Conferences and Journals

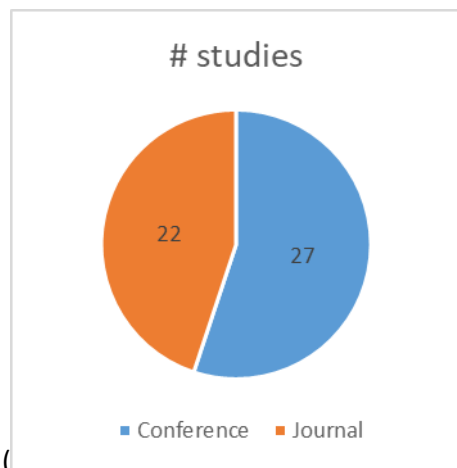


Figure 6). Table 17 shows the venues with more than one study in our set of selected papers. We have omitted the mention of venues appearing only once, however, more detailed information can be found in the DEF available online<sup>3</sup>.

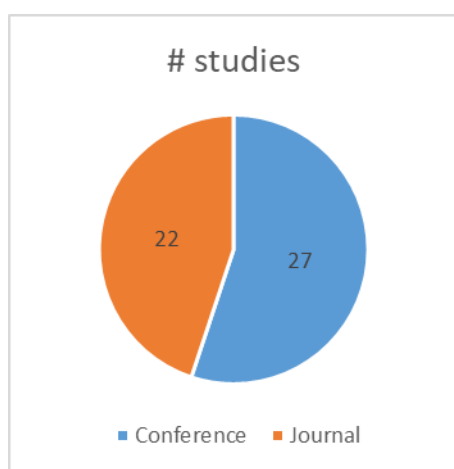


Figure 6 Publication venues

Table 17 Most mentioned venues

Venue	#papers
International Conference on Software Reuse (ICSR)	5
International Software Product Line Conference (SLPC)	5
IEEE Software	4
IEEE Transactions on Software Engineering	4
Journal of Systems and Software	4
Lecture Notes in Computer Science	3
International Conference on COTS-Based Software Systems	2
Empirical Software Engineering	2
Asia-Pacific Software Engineering Conference, APSEC	2

#### 4.2.2. PQ2: Publications per year

The number of publications per year varies widely in the range considered (1990-2016). With the available data, a trend cannot be predicted. The periods of greatest activity correspond to the years 1995, 2005 and 2015 (Figure 7).

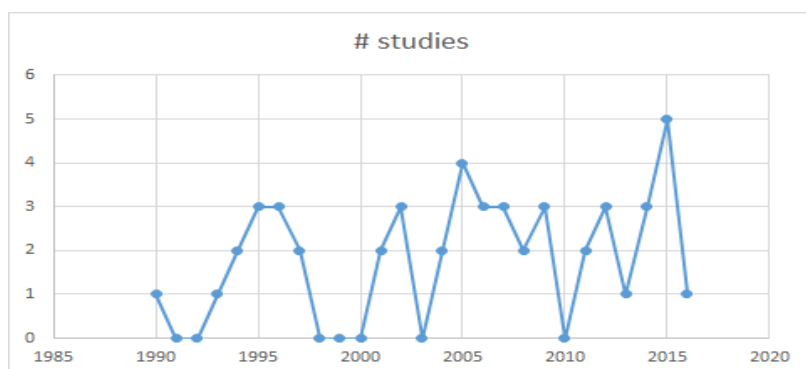


Figure 7 Evolution of publications

#### 4.2.3. PQ3: Active researchers

Sixteen authors have more than one work included in our set of selected primary works. Table 18 shows an ordered

<sup>3</sup><https://doi.org/10.6084/m9.figshare.5178148>

list of authors (by number of selected works). The data allow identifying groups of researchers who often publish collaborative works, for example: Conradi and Mohagheghi (S7 [39], S8 [40], S41 [72]); Gupta, Landre and Ronneberg (S8, S26 [57]); Ha and Sun (S3 [35], S39 [70]) and Knodel and Muthig (S17 [48], S34 [65]).

Table 18 Active researchers

Author	Selected primary works
Conradi, R	S7, S8, S26, S41
Mohagheghi, P.	S7, S8, S41
Frakes, W.	S4, S10
Goldin, L.	S2, S23
Griss, M.L.	S20, S29
Gupta, A.	S8, S26
Ha, W.	S3, S39
Knodel, J.	S17, S34
Kuflik, T.	S2, S28
Landre, E.	S8, S26
Muthig, D.	S17, S34
Ronneberg, H.	S8, S26
Succi, G.	S10, S42
Sun, H.	S3, S39
Weil, F.	S48, S49
Xie, M.	S3, S39

#### 4.2.4. PQ4: Researcher's affiliation (academic/industry)

Table 19 lists the most active institutions (by considering the author's affiliation). It is noteworthy the balance between industry and academia (38 authors from Academia versus 26 from Industry).

Table 19 Researcher's affiliation (Institutions)

Institution	Affiliation	# authors
The Norwegian University of Science and Technology	Academic	8
Fraunhofer Institute for Experimental Software Engineering	Academic	7
City University of Hong Kong	Academic	6
Motorola Global Software	Industry	6
Dipartimento di Informatica, Università di Genova	Academic	5
Department of Computer Science, NTNU	Academic	4
Freescale Semiconductor India Pvt Ltd.	Industry	4
Fujitsu Kyushu Network Technologies Limited	Industry	4
Hewlett-Packard Labs	Industry	4
Hitachi High-Technologies Corporation	Industry	4
Mälardalen University	Academic	4
Statoil KTJ/IT	Industry	4
Università di Bari	Academic	4

#### 4.2.5. PQ5: Active countries (author's affiliation)

To identify the most active countries, we took into consideration the affiliation of all the authors in the selected papers. USA (23.5%) and Norway (11.4%) are the most prolific countries in terms of author's affiliation.

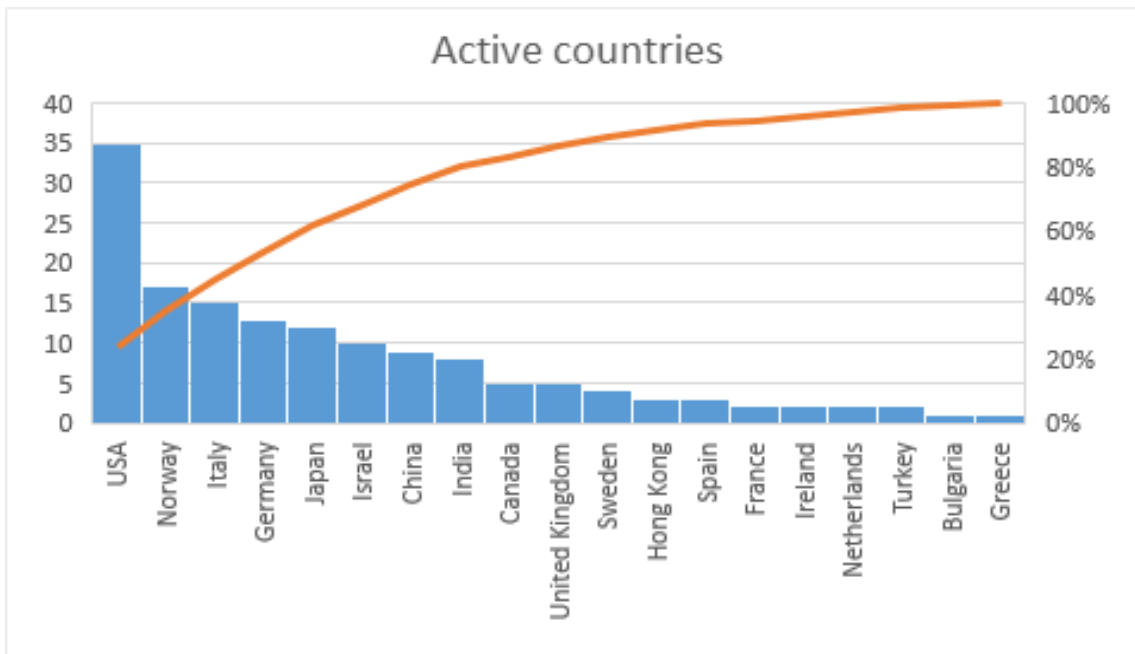


Figure 8 Active countries

### 4.3 Cross analysis of results

As a summary, Figure 9 shows the classification scheme and the output of this SMS. The map is in the form of a tree to improve readability. The tree shows the facets considered in the research questions of the SMS. For all facets, only the instances that are mentioned at least twice are included in the map. For example, in the “Reuse Benefits” facet, the benefits that were only mentioned once in the selected literature (such as “Automation” – see table 10) are not included in the tree. Therefore, this criterion for constructing the map best represents the available evidence.

The “Reuse Benefit” facet (RQ1) shows the 12 main identified benefits. The following facet, “Reuse Process” (RQ2), classified the four reported reuse processes. The third facet represents RQ3, containing eight different industry’s domains, plus a “Not reported” category. The “Type of reuse” (RQ4) also includes the “Not reported” category.

Three research types (RQ5) were identified. Finally, we did not include RQ6 (validity threats) as a classification facet, since we believe it would add little value to of the mapping representation.

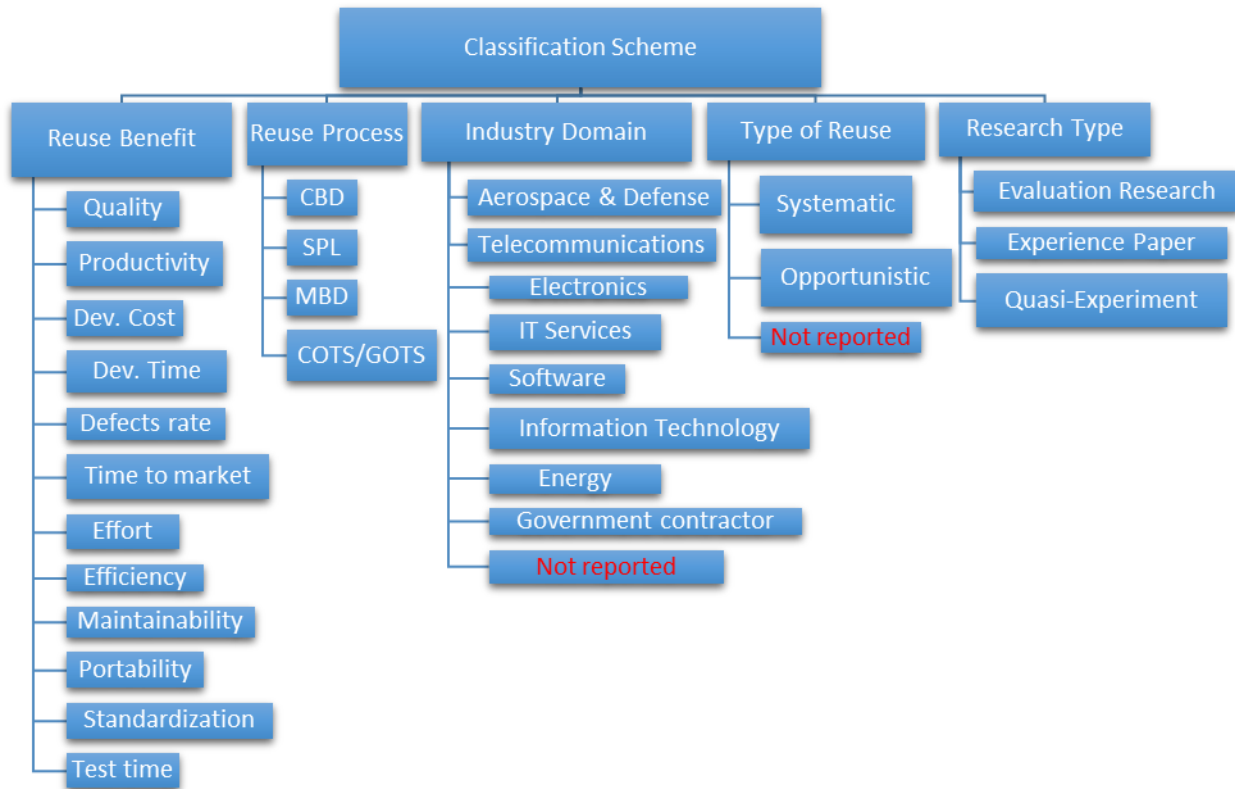


Figure 9 Summary of Results for the research space (the classification scheme)

#### 4.3.1. Reuse processes vs benefits

The observed reuse processes have impact on the benefits reported. According to our findings (see section 4.1.2) CBD is the most reported reuse process, followed by SPL. These two reuse processes seem to be the most comprehensive in terms of their capacity to show results across the different reported benefits (Table 20).

Table 20 Reuse processes and their impact on benefits

Reported Benefit	Reuse process			
	CBD	SPL	MBD	COTS/GOTS
Quality	S1, S5, S7, S8, S10, S14, S19, S25, S26, S37, S39	S16, S35, S43, S46	S15, S47-S49	
Productivity	S1, S5, S10, S14, S19, S25, S37, S39, S44	S9, S33, S43, S46	S13, S15, S22, S48, S49	
Development-cost	S1, S5, S8, S14, S39, S44	S16, S28, S34, S35	S45	
Development-time	S1, S8, S39	S18, S34, S35	S32	S30

Although the number of papers analyzed is small, the coverage of the CBD process can be highlighted with respect to the reported benefits. From the 16 papers dealing with the application of CBD, 11 papers reported an increase in quality and 9 reported a productivity improvement (or 12, if we take the reduction of development time as a synonym for improved productivity). Second, but far from CBD, SPLs show uniform coverage of the four top benefits (4 mentions of each benefit). These results suggest that CBD and SPL are the reuse processes that are most likely to result in benefits transferred to industry. In addition, the most frequently obtained benefits will be quality and productivity increments.

#### 4.3.2. Reuse processes vs. industry's domain

We limited the discussion here to the five most mentioned industry domains. We have found that the CBD and SPL reuse processes have been used in all reported domains, whereas COTS/GOTS is only referred to in the aerospace and defense domain. The use of COTS/GOTS in the aerospace and defense domain responds to the strict security and standardization requirements imposed by contractors. In that sense, our results coincide with other reports in the literature.

Table 21 Reuse processes and their relation to industry's domain

Industry domain	Reuse process			
	CBD	SPL	MBD	COTS/GOTS
Aerospace & Defense	S1, S37, S40	S9, S11		S30, S31
Telecommunications	S5, S7, S12, S41, S44	S17	S48, S49	
Electronics	S1, S25	S18, S34	S38	
IT Services	S1, S19	S16, S46	S13, S15, S47	
Software	S39	S2, S35		

#### 4.3.3. Benefits vs. type of reuse

Another interesting cross analysis is to observe which benefits have been obtained with the different approaches to reuse (Systematic or opportunistic). The difference between this analysis and the analysis in section 4.3.1 lies in the intent toward the reuse. As mentioned in the introduction, we rely on the standard definition of systematic reuse as “the practice of reuse according to a consistent, repeatable process”. As a result, for instance with S14, reported a systematic increase in Quality, but, though the effects were also measureable in Development-cost and productivity (see Table 20), in this analysis, only explicit intent was considered as “systematic”. Table 22 presents how the top identified benefits were achieved by a systematic reuse approach.

Table 22 Benefits vs type of reuse

Reported Benefit	Reuse approach		Not Reported
	Systematic/Planned	Opportunistic/ad-hoc	
Quality	S14, S16, S20, S21, S27, S29, S35, S47	S42	S7, S9
Productivity	S1, S13, S14, S33, S36, S37, S39, S43, S46, S48, S49		S22
Development-cost	S1, S5, S8, S28, S45	S2	S24
Development-time	S1, S4, S18, S32	S1	

#### 4.3.4. Semantics of the identified benefits

After reviewing the results, in particular with respect to the identified benefits, we conclude that research has been biased towards the first level of project management results. In [84], the author argues that there can be several levels of criteria for evaluating the success of software development projects. The first level deals with efficiency and performance of the execution of the project. This is, the effective management of resources in the six dimensions of project management as identified by the Project Management Body of Knowledge<sup>4</sup> (scope, cost, quality, resources, risks and time).

Of these six areas, we were able to identify benefits in the following four:

- Scope and Time: If we assume a definition of productivity as “unit of input (scope) times unit of time”, then the following benefits can be assigned to these two management areas: Productivity, Development time, time to market and test time (36.72% of the benefit's mentions).
- Quality: we included the studies reporting Quality, Defects rate, Portability, Maintainability, Standardization, Automation, Platform independence, Reliability, Upgradeability, Stability and Safety (39.06%).
- Cost: this area included studies reporting Development cost (16 mentions, 12.50%).

In summary, we argue that we did not identify benefits of reuse that aim at improving resource management nor risk management. Furthermore, there is limited evidence of benefits that can be assigned to the higher levels of success criteria presented by Dalcher [84].

## 5. Validity evaluation

We considered the four types of possible threats to validity mentioned in [13] during the design of the SMS protocol,

<sup>4</sup><https://www.pmi.org/pmbok-guide-standards>

and mitigated them as follows:

- Descriptive validity: *“the extent to which observations are described accurately and objectively”* [13]. To reduce this threat, we designed a DEF that objectively quantifies the data extraction process. By allowing for public access to this DEF, external reviewers can access it and revise the original data. Therefore, we considered this threat to be controlled. The DEF is available online at <https://doi.org/10.6084/m9.figshare.5178148>.
- Theoretical validity: *“the ability of being able to capture what we intend to capture”* [13], where we considered two activities:
  - Study identification/sampling: (missed studies). To reduce this threat, we applied four search strategies: automatic search, backward and forward snowballing, a manual search of publications profile for most cited authors and a manual search of works published in well-known international conferences. As each search strategy introduces some type of bias, we argue that this approach leads us to a fair sample of sources about a research topic.
  - Data extraction and classification: we apply the protocol detailed in the previous section.
- Interpretive validity: *“achieved when the conclusions drawn are reasonable given the data”* [13]. We applied the same strategy as that used for data extraction, dividing the work in two halves and having pairs of reviewers performing overlapping analysis. To deal with disagreements, we applied the criteria described in Table 8.
- Repeatability: *“requires detailed reporting of the research process”* [13]. We offered detailed reporting of the research process and all the data gathered during our study. The detailed protocol [26] and the data extraction form are available on the Internet for public access.

In addition, the relatively low number of selected works raise questions about results could be generalized. In particular, after having reviewed our process, we believe that despite the relatively large body of knowledge on software reuse, the reported results based on solid evidence are by far scarce. This fact is enhanced by the focus of this research (limited to industrial contexts).

Finally, we invited two external researchers to review and validate our research protocol in order to obtain feedback and opportunities for improvement. The output of this external review was a template, adapted from [13], to conduct a self-assessment of the work done to “systematically develop” this SMS. The template can be used as a self-evaluation tool, to help authors check whether their methodology has been done correctly (see Appendix 4).

## 6. Discussion

This section presents a few reflections on the current state of research in software reuse, given the evidence and data from the previous Results section. We address the implications that these results may have for practitioners and researchers, although the implications related to the researchers are developed in detail in the Conclusions and future work section.

Studies reporting data from real-world projects are surprisingly few. A large number of papers are limited to the description of the software reuse. Other studies present “solution proposals” to implement reuse processes, tools or techniques in organizations, but lack evaluations of their outcomes in real-world settings.

Although our paper selection criteria are able to filter papers of poor quality (i.e., lack of data, absence of solid research method, or without an examination of the validity threats), it is not sufficient for a complete study of rigor relevance. In such a case, we must use the criteria proposed by Ivarsson and Gorschek [85]. However, we believe that the three following conditions are the most helpful for selecting useful papers for practitioners to review:

- (a) Obsolescence (year of publication, we prefer more recent studies);
- (b) Context (we prefer Industrial over Academic studies)
- (c) Report of threats to validity (we prefer studies that report threats).



The application of this criteria results in the selection of the sources described in Table 23.

*Table 23 Recommended references from the set of selected papers*

PaperIDs	Topic	Insights
S10	Benefit: Quality	Direct relationship between amount of reuse and quality.
S23	Benefit: Productivity	Reduced Time-to-Market. Aerospace Industries.
S2	Benefit: Cost	A detailed model to compare reuse alternatives and compute Cost.
S11	Process: CBD	Embedded versus non-embedded in aerospace industries.
S18	Process: SPL	Reduced development time and effort. The study included researchers and practitioners (Already commented in section 4.1.1)
S45	Process: MBD	Lessons learnt from the adoption of MBD in railway signalling.
S30, S31	Process: COTS/GOTS	Reduction in custom code by 50%, albeit high cost of training. Study at NASA (Already commented in section 4.1.2)

Based upon the findings, our favoured approach could be systematic software reuse, given the amount of papers mentioning it (78% of the papers selected). However, it is important to conduct a previous cost/benefit analysis to help inform about the level of "systematic reuse" to be integrated into the software development lifecycle.

The findings suggest that the four dimensions of software project management (scope, cost, time and quality) are present in the reported benefits, but more research is needed to understand how benefits stem from the application of reuse activities. For instance, CBD is the reuse process that reported most benefits; however, we did not find any evidence that CBD would reduce maintenance time. Likewise, we found no evidence of the impact of SPL on reliability; does this mean that a SPL cannot be designed to achieve reliable products?

A noteworthy aspect is the scarce presence of studies having data about costs. Although the reduction of the development time is mentioned, it is not monetized. This lack of monetization also occurs with respect to the benefits of quality increases and the reduction of time-to-market. It is necessary to quantify these benefits monetarily, since investments in software reuse have to compete with other investments within an organization.

There are a broad set of possibilities for further research in software reuse. All studies insist on the limited availability of objective data, extracted from projects in real-world contexts. Few studies have been done on potential methods to obtain this data, or the development of software tools that can automate, to some extent, the process of collecting and analyzing that data. In addition, the relationship between the type of reuse processes and the type of benefits that can be obtained have not been thoroughly studied. Currently, there are no standard metrics available to compare data from different studies. There are some proposals for cost models, but little progress has been made in standardization.

Another issue that brings opportunities for deeper investigation is that of research methods. A significant number of studies rely on surveys that have not been well designed, or whose data were not analyzed with the appropriate statistics. There is a lack of experiments, as well as explicit rules for carrying out case studies with a solid methodological base. The next section delves into these, and other, research gaps.

## 7. Conclusions and future work

This paper has presented a systematic mapping study on the benefits that were reported after the application of software reuse processes. The systematic mapping study was conducted following the guidelines by [13]. For reproducibility of this study, care was taken to document the steps of the research protocol that is publicly available [26]. Four complementary search strategies were used: a) automatic search conducted in four publishers and indexers databases, b) backwards and forwards snowballing, c) manual search of known conferences and d) manual search of publication profiles from most cited authors. The search strategies retrieved 2,413 papers and a set of 49 primary works were selected after the application of the inclusion/exclusion criteria. This relatively low number of selected works raises questions as to whether the results can be objectively generalized. In particular, after having

reviewed our process, we believe that in spite of the relatively large body of knowledge on software reuse, the reported results based on solid evidence are by far scarce. This fact is enhanced by the focus of this research (limited to industrial contexts).

Our work identified 26 benefits obtained by the application of 13 reuse processes in 15 industrial domains. The most reported benefit was *quality increase*, the most frequent process was component-based development and the most cited industry domain was aerospace & defense. Furthermore, our results show that 78% of the selected works (38 out of 49) report benefits that have been achieved by applying systematic reuse. We believe that this result strengthens the motivation expressed by the standard *“Software reuse entails capitalizing on existing software and systems to create new products, according to a consistent, repeatable process”* [7].

Thirty-one works (63%) applied evaluation research as defined in [28], either using case studies (50%) or surveys (12%) as a research method. This type of research implies use in practice and an empirical evaluation (see Table 3). Only 36% of the works were experience reports, in which the evaluation mechanism is not always reported. We believe that these percentages reflect the high scientific quality of the set of papers we selected.

Another interesting result of our study is the diversity of terms used to report quality. Terms such as defects rate, reliability, maintainability, security, upgradeability, customer satisfaction, safety or stability are mentioned as quality's synonyms.

Some open issues for future research could include:

- To conduct a systematic literature review to deepen the knowledge about the processes of reuse and how the benefits are transferred.
- The number of selected works for this study is small. This provides an opportunity to investigate the development of simpler protocols that can be used in industry to gather relevant data, such as return on investment, while applying rigorous methods.
- Use a quality model, such as the one proposed in the standard SQuaRE [24], to integrate the reported benefits and relate them to specific reuse processes.
- Develop and validate a model suitable to link reuse benefits to economic values (strategic or financial).
- Apply other research methods to advance theory on software reuse. The most widely used research method was reported in our selected sample of sources was the case study (25 of 49). This method includes validation of hypotheses and provides some assurance of the validity of the results, however, the second most common type of work was Experience Report (18 of 49), in which no validation of the results is performed. To advance the theory of software reuse beyond conjectures and hypotheses, other types of research methods are needed. However, we understand the difficulties and feasibility issues in designing reuse related experiments. In particular, the selected case studies could be further evaluated by a process similar to the reported in [86], which would allow us to continue exploring which reuse benefits have been transferred to the industry.

We believe this study can be of interest to practitioners, to get an overview of the current state of the art and identify reuse strategies that offered a higher potential for gains in a given context. On the other hand, researchers can benefit from the results obtained by this SMS, as a help to decide on the research areas that deserve further study, or the most appropriate research methods and reporting tips to deal with current issues, such as the lack of empirical data.

## Acknowledgments

We would like to thank the following people: Ania Cravero and Marcela Genero who reviewed the protocol and offered interesting insights about avoiding validity threats; Iván García and Carla Lenínca who helped in refining the automatic search strategy; Laura Zeligueta who participated in the initial paper selection process, and David Olivieri who struggled with the subtleties of the English language.

## References

- [1] W. Tracz, COTS myths and other lessons learned in component-based software development, in: G.T.H. and W.T. Councill (Ed.), *Compon. Based Softw. Eng. - Putt. Pieces Together*, Addison Wesley Professional, 2001: pp. 99–111.
- [2] M.A. Rothenberger, K.J. Dooley, A performance measure for software reuse projects, *Decis. Sci.* 30 (1999) 1131–1153.
- [3] M.D. McIlroy, J. Buxton, P. Naur, B. Randell, Mass-Produced Software Components, in: *Proc. 1st Int. Conf. Softw. Eng.*, Garmisch Pattenkirchen, Germany, 1968: pp. 88–98.
- [4] C.W. Krueger, Software reuse, *ACM Comput. Surv.* 24 (1992) 131–183.
- [5] W.B. Frakes, K. Kang, Software reuse research: Status and future, *IEEE Trans. Softw. Eng.* 31 (2005) 529–536.
- [6] P. Mohagheghi, R.R. Conradi, Quality, productivity and economic benefits of software reuse: A review of industrial studies, *Empir. Softw. Eng.* 12 (2007) 471–516. doi:10.1007/s10664-007-9040-x.
- [7] IEEE Standard, 1517-2010 - IEEE Standard for Information Technology--System and Software Life Cycle Processes--Reuse Processes, 2010.
- [8] E.S. De Almeida, A. Alvaro, D. Lucrédio, V.C. Garcia, S.R. de Lemos Meira, A survey on software reuse processes, in: *Inf. Reuse Integr. Conf*, 2005. IRI-2005 IEEE Int. Conf. On., 2005: pp. 66–71.
- [9] J.L. Barros-Justo, F.B. V Benitti, A.L. Cravero-Leal, Software patterns and requirements engineering activities in real-world settings: a systematic mapping study, *Comput. Stand. Interfaces.* (2017).
- [10] V. Bauer, Facts and fallacies of reuse in practice, in: 2013 17th Eur. Conf. Softw. Maint. Reengineering, IEEE, 2013: pp. 431–434. doi:10.1109/CSMR.2013.65.
- [11] B. Deniz, S. Bilgen, An Empirical Study of Software Reuse and Quality in an Industrial Setting, in: *Comput. Sci. Its Appl.* 2014, Springer, 2014: pp. 508–523.
- [12] D. Lucrédio, K. dos Santos Brito, A. Alvaro, V.C. Garcia, E.S. de Almeida, R.P. de Mattos Fortes, S.L. Meira, Software reuse: The Brazilian industry scenario, 81 (2008) 996–1013. doi:10.1016/j.jss.2007.08.036.
- [13] K. Petersen, S. Vakkalanka, L. Kuzniarz, Guidelines for conducting systematic mapping studies in software engineering: An update, *Inf. Softw. Technol.* 64 (2015) 1–18. doi:10.1016/j.infsof.2015.03.007.
- [14] C. Wohlin, Guidelines for snowballing in systematic literature studies and a replication in software engineering, in: 18th Int. Conf. Eval. Assess. Softw. Eng. (EASE 2014), 2014: p. 38. doi:10.1145/2601248.2601268.
- [15] P. Mohagheghi, V. Dehlen, Where Is the Proof - A Review of Experiences from Applying MDE in Industry, in: *Eur. Conf. Model Driven Archit. Appl.*, 2008: pp. 432–443.
- [16] S. Montagud, S. Abrahão, E. Insfran, A systematic review of quality attributes and measures for software product lines, *Softw. Qual. J.* 20 (2012) 425–486. doi:10.1007/s11219-011-9146-7.
- [17] H. Kahtan, N.A. Bakar, R. Nordin, Reviewing the challenges of security features in component based software development models, 2012 IEEE Symp. E-Learning, E-Management E-Services, IS3e 2012. (2012) 105–110. doi:10.1109/IS3e.2012.6414955.
- [18] V. Myllärniemi, M. Raatikainen, T. Männistö, A Systematically Conducted Literature Review: Quality Attribute Variability in Software Product Lines, *Proc. 16th Int. Softw. Prod. Line Conf. - Vol. 1.* 1 (2012) 41–45. doi:10.1145/2362536.2362546.
- [19] L.R. Soares, P. Potena, I. do C. Machado, I. Crnkovic, E.S. De Almeida, Analysis of Non-functional Properties in Software Product Lines: A Systematic Review, 2014 40th EUROMICRO Conf. Softw. Eng. Adv. Appl. (2014)

328–335. doi:10.1109/seaa.2014.48.

- [20] D. Flemstrom, D. Sundmark, W. Afzal, Vertical Test Reuse for Embedded Systems: A Systematic Mapping Study, 2015 41st Euromicro Conf. Softw. Eng. Adv. Appl. (2015) 317–324. doi:10.1109/SEAA.2015.46.
- [21] J. Varnell-sarjeant, A.A. Andrews, Comparing Reuse Strategies in Different Development Environments, in: *Adv. Comput.*, 1st ed., Elsevier {BV}, 2015: pp. 1–47. doi:10.1016/bs.adcom.2014.10.002.
- [22] G. Scanniello, C. Gravino, M. Risi, G. Tortora, A controlled experiment for assessing the contribution of design pattern documentation on software maintenance, in: *Proc. 2010 ACM-IEEE Int. Symp. Empir. Softw. Eng. Meas.*, 2010: p. 52.
- [23] M. Goulão, V. Amaral, M. Mernik, Quality in model-driven engineering: a tertiary study, *Softw. Qual. J.* 24 (2016) 601–633. doi:10.1007/s11219-016-9324-8.
- [24] ISO/IEC 25010:2011 Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models, 2011.
- [25] M. Irshad, R. Torkar, K. Petersen, W. Afzal, Capturing cost avoidance through reuse, *Proc. 20th Int. Conf. Eval. Assess. Softw. Eng. - EASE '16.* (2016) 1–12. doi:10.1145/2915970.2915989.
- [26] J. Barros-Justo, F. Pinciroli, S. Matalonga, M. Gonzalez, N. Martinez Araujo, Systematic Mapping Protocol: Have Systematic Reuse Benefits Been Transferred to Real-world Settings?, *ArXiv E-Prints.* (2016).
- [27] Global Industry Classification Standard (GICS) | Thomson Reuters, (n.d.). <https://www.thomsonreuters.com/en/resources/third-party-restrictions/gics.html> (accessed June 2, 2017).
- [28] R. Wieringa, N. Maiden, N. Mead, C. Rolland, Requirements engineering paper classification and evaluation criteria: A proposal and a discussion, *Requir. Eng.* 11 (2006) 102–107. doi:10.1007/s00766-005-0021-6.
- [29] B. Kitchenham, P. Brereton, M. Turner, M. Niazi, S. Linkman, R. Pretorius, D. Budgen, The impact of limited search procedures for systematic literature reviews - A participant-observer case study, in: *3rd Int. Symp. Empir. Softw. Eng. Meas.*, 2009: pp. 336–345. doi:10.1109/ESEM.2009.5314238.
- [30] B. Kitchenham, P. Brereton, Z. Li, D. Budgen, A. Burn, Repeatability of systematic literature reviews, in: *Eval. Assess. Softw. Eng. (EASE 2011), 15th Annu. Conf.*, 2011: pp. 46–55.
- [31] S. MacDonell, M. Shepperd, B. Kitchenham, E. Mendes, How Reliable Are Systematic Reviews in Empirical Software Engineering?, *IEEE Trans. Softw. Eng.* 36 (2010) 676–687. doi:10.1109/TSE.2010.28.
- [32] B. Kitchenham, S. Charters, Guidelines for performing Systematic Literature Reviews in Software Engineering. Version 2.3, 2007. doi:10.1145/1134285.1134500.
- [33] A.J. Incorvaia, M. Street, A.M. Davis, R.E. Fairley, Case Studies in Software Reuse', *Syst. Eng.* (1990) 301–306.
- [34] A. Tomer, L. Goldin, T. Kuflik, E. Kimchi, S.R. Schach, Evaluating Software Reuse Alternatives: A Model and Its Application to an Industrial Case Study, *IEEE Trans. Softw. Eng.* 30 (2004) 601–612.
- [35] W. Ha, H. Sun, M. Xie, Reuse of embedded software in small and medium enterprises, in: *2012 IEEE 6th Int. Conf. Manag. Innov. Technol. ICMIT 2012*, 2012: pp. 394–399.
- [36] C. Hollenbach, W. Frakes, Software process reuse in an industrial setting, *Proc. Fourth IEEE Int. Conf. Softw. Reuse.* (1996) 22–30. doi:10.1109/ICSR.1996.496110.
- [37] R. Joos, Software reuse at Motorola, *IEEE Softw.* 11 (1994) 42–47.
- [38] A.S. Orrego, G.E. Mundy, A study of software reuse in NASA legacy systems, *Innov. Syst. Softw. Eng.* 3 (2007) 167–180.
- [39] P. Mohagheghi, R. Conradi, An empirical investigation of software reuse benefits in a large telecom product, *ACM Trans. Softw. Eng. Methodol.* 17 (2008) 1–31. doi:10.1145/1363102.1363104.

- [40] O.P.N. Slyngstad, A. Gupta, R. Conradi, P. Mohagheghi, H. Rønneberg, E. Landre, An empirical study of developers views on software reuse in statoil ASA, in: ISESE'06 - Proc. 5th ACM-IEEE Int. Symp. Empir. Softw. Eng., 2006: pp. 242–251. doi:10.1145/1159733.1159770.
- [41] W.B. Frakes, G. Succi, An industrial study of reuse, quality and productivity, *J. Syst. Softw.* 57 (2001) 99–106. doi:10.1016/S0164-1212(00)00121-7.
- [42] J. Varnell-Sarjeant, A. Amschler Andrews, J. Lucente, A. Stefik, Comparing development approaches and reuse strategies: An empirical evaluation of developer views from the aerospace industry, *Inf. Softw. Technol.* 61 (2015) 71–92. doi:10.1016/j.infsof.2015.01.002.
- [43] M. Ramachandran, W. Fleischer, Design for Large Scale Software Reuse: An Industrial Case Study, in: 4th Int. Conf. Softw. Reuse, 1996: pp. 104–111. [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=496118](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=496118).
- [44] D. Kamma, S.K. G, Effect of Model Based Software Development on Productivity of Enhancement Tasks -- An Industrial Study, in: 2014 21st Asia-Pacific Softw. Eng. Conf., IEEE, 2014: pp. 71–77. doi:10.1109/APSEC.2014.20.
- [45] W.C. Lim, Effects of reuse on quality, productivity, and economics, *IEEE Softw.* 11 (1994) 23–30. doi:10.1109/52.311048.
- [46] W. Heijstek, M.R. V Chaudron, Empirical investigations of model size, complexity and effort in a large scale, distributed model driven development process, *Conf. Proc. EUROMICRO.* (2009) 113–120. doi:10.1109/SEAA.2009.70.
- [47] G. Quilty, M.Ó. Cinneide, Experiences with Software Product Line Development in Risk Management Software, in: 2011 15th Int. Softw. Prod. Line Conf., IEEE, 2011: pp. 251–260. doi:10.1109/SPLC.2011.30.
- [48] H.J. Beyer, D. Hein, C. Schitter, J. Knodel, D. Muthig, M. Naab, Introducing architecture-centric reuse into a small development organization, *Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics).* 5030 LNCS (2008) 1–13. doi:10.1007/978-3-540-68073-4\_1.
- [49] D. Sellier, M. Mannion, G. Benguria, G. Urchegui, Introducing software product line engineering for metal processing lines in a small to medium enterprise, *Proc. - 11th Int. Softw. Prod. Line Conf. SPLC 2007.* (2007) 54–60. doi:10.1109/SPLINE.2007.4339255.
- [50] E. Henry, B. Faller, Large-Scale Industrial Reuse to Reduce Cost and Cycle Time, *IEEE Softw.* 12 (1995) 47–53. doi:10.1109/52.406756.
- [51] M.L. Griss, M. Wosser, Making reuse work at Hewlett-Packard, *IEEE Softw.* 12 (1995) 105–107. doi:10.1109/52.363160.
- [52] G. Butler, Quality and reuse in industrial software engineering, *Softw. Eng. Conf. 1997. Asia Pacific Int. Comput. Sci. Conf. 1997. APSEC '97 ICSC '97. Proc.* (1997) 3–12. doi:10.1109/APSEC.1997.640156.
- [53] M. Torchiano, F. Tomassetti, F. Ricca, A. Tiso, G. Reggio, Relevance, benefits, and problems of software modelling and model driven techniques - A survey in the Italian industry, *J. Syst. Softw.* 86 (2013) 2110–2126. doi:10.1016/j.jss.2013.03.084.
- [54] L. Goldin, D.M. Berry, Reuse of requirements reduced time to market at one industrial shop: a case study, *Requir. Eng.* 20 (2015) 23–44. doi:10.1007/s00766-013-0182-7.
- [55] R. Land, D. Sundmark, F. Lüders, I. Krasteva, A. Causevic, Reuse with Software Components - A Survey of Industrial State of Practice, in: *Int. Conf. Softw. Reuse, 2009: pp. 150–159.* doi:10.1007/3-540-44947-7\_30.
- [56] R. Srivastava, N. Mudgil, G. Gupta, H. Mondal, SoC time to market improvement through device driver reuse: An industrial experience, *Proc. - 2012 Int. Symp. Electron. Syst. Des. ISED.* (2012) 56–61. doi:10.1109/ISED.2012.61.
- [57] A. Gupta, J. Li, R. Conradi, H. Rønneberg, E. Landre, A case study comparing defect profiles of a reused framework and of applications reusing it, *Empir. Softw. Eng.* 14 (2009) 227–255.

- [58] M.T. Baldassarre, A. Bianchi, D. Caivano, G. Visaggio, An industrial case study on reuse oriented development, in: IEEE Int. Conf. Softw. Maintenance, ICSM, 2005: pp. 283–294.
- [59] S. Morad, T. Kuflik, Conventional and Open Source Software Reuse at Orbotech - An Industrial Experience, IEEE Int. Conf. Softw. - Sci. Technol. Eng. 2005 (2005) 110–117. doi:10.1109/SWSTE.2005.11.
- [60] M.L. Griss, Software reuse: From library to factory, IBM Syst. J. 32 (1993) 548–566. doi:10.1147/sj.324.0548.
- [61] T. Pfarr, J.E. Reis, The integration of COTS/GOTS within NASA's HST command and control system., in: Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), 2002: pp. 209–221.
- [62] S.K. Semancik, A.M. Conger, The Standard Autonomous File Server, a Customized, O -The-Shelf Success Story, (2002) 234–244.
- [63] von A. Mayrhauser, M. Shumway, P. Ocken, R. Mraz, On Domain Models for System Testing, (1996) 114–123. doi:10.1016/S0065-2458(08)60285-5.
- [64] R. Kodama, J. Shimabukuro, Y. Takagi, S. Koizumi, S. Tano, Experiences with Commonality Control Procedures to Develop Clinical Instrument System, Proc. 18th Int. Softw. Prod. Line Conf. - Vol. 1. (2014) 254–263. doi:10.1145/2648511.2648540.
- [65] R. Kolb, I. John, J. Knodel, D. Muthig, U. Haury, G. Meier, Experiences with product line development of embedded systems at testo AG, Proc. - 10th Int. Softw. Prod. Line Conf. SPLC 2006. (2006) 172–181. doi:10.1109/SPLINE.2006.1691589.
- [66] J. Otsuka, K. Kawarabata, T. Iwasaki, M. Uchiba, T. Nakanishi, K. Hisazumi, Small inexpensive core asset construction for large gainful product line development: Developing a communication system firmware product line, in: ACM Int. Conf. Proceeding Ser., 2011. doi:10.1145/2019136.2019159.
- [67] S. Thakur, H. Singh, FDRD: Feature driven reuse development process model, Proc. 2014 IEEE Int. Conf. Adv. Commun. Control Comput. Technol. ICACCCT 2014. (2015) 1593–1598. doi:10.1109/ICACCCT.2014.7019376.
- [68] R.W. Selby, Enabling reuse-based software development of large-scale systems, IEEE Trans. Softw. Eng. 31 (2005) 495–510. doi:10.1109/TSE.2005.69.
- [69] M.J. Moreno-Lizaranzu, F. Cuesta, A framework and architecture for rapid software development: a success story, Empir. Softw. Eng. 20 (2014) 1456–1485. doi:10.1007/s10664-014-9320-1.
- [70] H. Sun, W. Ha, M. Xie, J. Huang, Modularity's impact on the quality and productivity of embedded software development: A case study in a Hong Kong company, Total Qual. Manag. 26 (2015) 1188–1201. doi:10.1080/14783363.2014.920179.
- [71] W. Thomas, A. Delis, V. Basili, An analysis of errors in a reuse-oriented development environment, J. Syst. Softw. 38 (1997) 211–224. doi:10.1016/S0164-1212(96)00152-5.
- [72] P. Mohagheghi, R. Conradi, O.M. Killi, H. Schwarz, An Empirical Study of Software Reuse vs. Defect-Density and Stability, in: Proc. 26th Int. Conf. Softw. Eng., IEEE Computer Society Press, 2004: pp. 282–292.
- [73] G. Succi, L. Benedicenti, T. Vernazza, Analysis of the effects of software reuse on customer satisfaction in an RPG environment, IEEE Trans. Softw. Eng. 27 (2001) 473–479. doi:10.1109/32.9780734038937.
- [74] M. Morisio, D. Romano, I. Stamelos, Quality , Productivity , and Learning in Framework-Based Development : an exploratory case study, IEEE Trans. Softw. Eng. 28 (2002) 876–888. doi:10.1109/TSE.2002.1033227.
- [75] S. Isoda, Experiences of a software reuse project, J. Syst. Softw. 30 (1995) 171–186. doi:10.1016/0164-1212(94)00132-7.
- [76] A. Ferrari, A. Fantechi, S. Gnesi, Lessons learnt from the adoption of formal model-based development, Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 7226 LNCS (2012) 24–38. doi:10.1007/978-3-642-28891-3\_5.
- [77] F. Mao, X. Cai, B. Shen, B. Jin, Operational Pattern Based Code Generation For Management Information

System : An Industrial Case Study, in: IEEE (Ed.), Softw. Eng. Artif. Intell. Netw. Parallel/Distributed Comput. (SNPD), 2016 17th IEEE/ACIS Int. Conf., 2016: pp. 425–430.

- [78] D. Shartz, M. Kazakov, Y. Shaham-Gafni, Adopting Model Driven Development in a Large Financial Organization, Model Driven Archit. Found. Appl. (2007) 172–183. doi:10.1007/978-3-540-72901-3\_13.
- [79] P. Baker, S. Loh, F. Weil, Model-Driven Engineering in a large industrial context—Motorola case study, Model Driven Eng. Lang. Syst. (2005) 476–491. doi:10.1007/11557432\_36.
- [80] T. Weigert, Practical Experiences in Using Model-Driven Engineering to Develop Trustworthy Computing Systems., in: Sens. Networks, Ubiquitous, Trust. Comput. 2006. IEEE Int. Conf., 2006: pp. 8–16. doi:10.1109/SUTC.2006.1636178.
- [81] I. Crnkovic, M.P.H. Larsson, Building reliable component-based software systems, Artech House, 2002.
- [82] International Organization for Standardization, ISO/IEC 26550:2015 Software and systems engineering -- Reference model for product line engineering and management, Stand. Cat. (2015). <https://www.iso.org/standard/69529.html>.
- [83] W.C. Lim, Managing software reuse: a comprehensive guide to strategically reengineering the organization for reusable components, Prentice-Hall, Inc., 1998.
- [84] D. Dalcher, Rethinking Success in Software Projects: Looking Beyond the Failure Factors, in: Softw. Proj. Manag. a Chang. World, Springer Berlin Heidelberg, Berlin, Heidelberg, 2014: pp. 27–49. doi:10.1007/978-3-642-55035-5\_2.
- [85] M. Ivarsson, T. Gorschek, A method for evaluating rigor and industrial relevance of technology evaluations, Empir. Softw. Eng. 16 (2011) 365–395. doi:10.1007/s10664-010-9146-4.
- [86] S. Stavru, A critical examination of recent industrial surveys on agile method usage, J. Syst. Softw. 94 (2014) 87–97. doi:10.1016/J.JSS.2014.03.041.

## Appendices

### 1. Selected (included) studies

S1	A.J. Incorvaia, M. Street, A.M. Davis, R.E. Fairley, Case Studies in Software Reuse ', Syst. Eng. (1990) 301–306.
S2	A. Tomer, L. Goldin, T. Kuflik, E. Kimchi, S.R. Schach, Evaluating Software Reuse Alternatives: A Model and Its Application to an Industrial Case Study, IEEE Trans. Softw. Eng. 30 (2004) 601–612.
S3	W. Ha, H. Sun, M. Xie, Reuse of embedded software in small and medium enterprises, in: 2012 IEEE 6th Int. Conf. Manag. Innov. Technol. ICMIT 2012, 2012: pp. 394–399..
S4	C. Hollenbach, W. Frakes, Software process reuse in an industrial setting, Proc. Fourth IEEE Int. Conf. Softw. Reuse. (1996) 22–30. doi:10.1109/ICSR.1996.496110.
S5	R. Joos, Software reuse at Motorola, IEEE Softw. 11 (1994) 42–47.
S6	A.S. Orrego, G.E. Mundy, A study of software reuse in NASA legacy systems, Innov. Syst. Softw. Eng. 3 (2007) 167–180.
S7	P. Mohagheghi, R. Conradi, An empirical investigation of software reuse benefits in a large telecom product, ACM Trans. Softw. Eng. Methodol. 17 (2008) 1–31. doi:10.1145/1363102.1363104.
S8	O. Slynstad, A. Gupta, R. Conradi, P. Mohagheghi, H. Rønneberg, E. Landre, An empirical study of developers views on software reuse in statoil ASA, in: ISESE'06 - Proc. 5th ACM-IEEE Int. Symp. Empir. Softw. Eng., 2006: pp. 242–251.
S9	B. Deniz, S. Bilgen, An Empirical Study of Software Reuse and Quality in an Industrial Setting, in: Comput. Sci. Its Appl. 2014, Springer, 2014: pp. 508–523.
S10	W. Frakes, G. Succi, An industrial study of reuse, quality and productivity, J. Syst. Softw. 57 (2001) 99–106.
S11	J. Varnell-Sarjeant, A. Amschler Andrews, J. Lucente, A. Stefik, Comparing development approaches and reuse strategies: An empirical evaluation of developer views from the aerospace industry, Inf. Softw. Technol. 61 (2015) 71–92. doi:10.1016/j.infsof.2015.01.002.
S12	M. Ramachandran, W. Fleischer, Design for Large Scale Software Reuse: An Industrial Case Study, in: 4th Int. Conf. Softw. Reuse, 1996:

	pp. 104–111.
S13	D. Kamma, S.K. G, Effect of Model Based Software Development on Productivity of Enhancement Tasks -- An Industrial Study, in: 2014 21st Asia-Pacific Softw. Eng. Conf., IEEE, 2014: pp. 71–77. doi:10.1109/APSEC.2014.20.
S14	W.C. Lim, Effects of reuse on quality, productivity, and economics, IEEE Softw. 11 (1994) 23–30.
S15	W. Heijstek, M.R. V Chaudron, Empirical investigations of model size, complexity and effort in a large scale, distributed model driven development process, Conf. Proc. EUROMICRO. (2009) 113–120. doi:10.1109/SEAA.2009.70.
S16	G. Quilty, M.Ó. Cinneide, Experiences with Software Product Line Development in Risk Management Software, in: 2011 15th Int. Softw. Prod. Line Conf., IEEE, 2011: pp. 251–260. doi:10.1109/SPLC.2011.30.
S17	H.J. Beyer, D. Hein, C. Schitter, J. Knodel, D. Muthig, M. Naab, Introducing architecture-centric reuse into a small development organization, Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 5030 LNCS (2008) 1–13. doi:10.1007/978-3-540-68073-4_1.
S18	D. Sellier, M. Mannion, G. Benguria, G. Urchegui, Introducing software product line engineering for metal processing lines in a small to medium enterprise, Proc. - 11th Int. Softw. Prod. Line Conf. SPLC 2007. (2007) 54–60. doi:10.1109/SPLINE.2007.4339255.
S19	E. Henry, B. Faller, Large-Scale Industrial Reuse to Reduce Cost and Cycle Time, IEEE Softw. 12 (1995) 47–53. doi:10.1109/52.406756.
S20	M.L. Griss, M. Wosser, Making reuse work at Hewlett-Packard, IEEE Softw. 12 (1995) 105–107. doi:10.1109/52.363160.
S21	G. Butler, Quality and reuse in industrial software engineering, Softw. Eng. Conf. 1997. Asia Pacific ... Int. Comput. Sci. Conf. 1997. APSEC '97 ICSC '97. Proc. (1997) 3–12. doi:10.1109/APSEC.1997.640156.
S22	M. Torchiano, F. Tomassetti, F. Ricca, A. Tiso, G. Reggio, Relevance, benefits, and problems of software modelling and model driven techniques - A survey in the Italian industry, J. Syst. Softw. 86 (2013) 2110–2126. doi:10.1016/j.jss.2013.03.084.
S23	L. Goldin, D.M. Berry, Reuse of requirements reduced time to market at one industrial shop: a case study, Requir. Eng. 20 (2015) 23–44. doi:10.1007/s00766-013-0182-7.
S24	R. Land, D. Sundmark, F. Lüders, I. Krasteva, A. Causevic, Reuse with Software Components - A Survey of Industrial State of Practice, in: Int. Conf. Softw. Reuse, 2009: pp. 150–159. doi:10.1007/3-540-44947-7_30.
S25	R. Srivastava, N. Mudgil, G. Gupta, H. Mondal, SoC time to market improvement through device driver reuse: An industrial experience, Proc. - 2012 Int. Symp. Electron. Syst. Des. ISED 2012. (2012) 56–61. doi:10.1109/ISED.2012.61.
S26	A.. Gupta, J.. Li, R.. Conradi, H.. Rønneberg, E.. Landre, A case study comparing defect profiles of a reused framework and of applications reusing it, Empir. Softw. Eng. 14 (2009) 227–255.
S27	M.T. Baldassarre, A. Bianchi, D. Caivano, G. Visaggio, An industrial case study on reuse oriented development, in: IEEE Int. Conf. Softw. Maintenance, ICSM, 2005: pp. 283–294.
S28	S. Morad, T. Kuflik, Conventional and Open Source Software Reuse at Orbotech - An Industrial Experience, IEEE Int. Conf. Softw. - Sci. Technol. Eng. 2005 (2005) 110–117. doi:10.1109/SWSTE.2005.11.
S29	M.L. Griss, Software reuse: From library to factory, IBM Syst. J. 32 (1993) 548–566. doi:10.1147/sj.324.0548.
S30	T. Pfarr, J.E. Reis, The integration of COTS/GOTS within NASA's HST command and control system., in: Int. Conf. COTS-Based Softw. Syst., 2002: pp. 209–221.
S31	S.K. Semancik, A.M. Conger, The Standard Autonomous File Server, a Customized, O -The-Shelf Success Story, (2002) 234–244.
S32	von A. Mayrhauser, M. Shumway, P. Ocken, R. Mraz, On Domain Models for System Testing, (1996) 114–123. doi:10.1016/S0065-2458(08)60285-5.
S33	R. Kodama, J. Shimabukuro, Y. Takagi, S. Koizumi, S. Tano, Experiences with Commonality Control Procedures to Develop Clinical Instrument System, Proc. 18th Int. Softw. Prod. Line Conf. - Vol. 1. (2014) 254–263. doi:10.1145/2648511.2648540.
S34	R. Kolb, I. John, J. Knodel, D. Muthig, U. Haury, G. Meier, Experiences with product line development of embedded systems at testo AG, Proc. - 10th Int. Softw. Prod. Line Conf. SPLC 2006. (2006) 172–181. doi:10.1109/SPLINE.2006.1691589.
S35	J. Otsuka, K. Kawarabata, T. Iwasaki, M. Uchiba, T. Nakanishi, K. Hisazumi, Small inexpensive core asset construction for large gainful product line development: Developing a communication system firmware product line, in: ACM Int. Conf. Proceeding Ser., 2011. doi:10.1145/2019136.2019159.
S36	S. Thakur, H. Singh, FDRD: Feature driven reuse development process model, Proc. 2014 IEEE Int. Conf. Adv. Commun. Control Comput. Technol. ICACCCT 2014. (2015) 1593–1598. doi:10.1109/ICACCCT.2014.7019376.
S37	R.W. Selby, Enabling reuse-based software development of large-scale systems, IEEE Trans. Softw. Eng. 31 (2005) 495–510. doi:10.1109/TSE.2005.69.
S38	M.J. Moreno-Lizaranzu, F. Cuesta, A framework and architecture for rapid software development: a success story, Empir. Softw. Eng. 20 (2014) 1456–1485. doi:10.1007/s10664-014-9320-1.



S39	H. Sun, W. Ha, M. Xie, J. Huang, Modularity's impact on the quality and productivity of embedded software development: A case study in a Hong Kong company, <i>Total Qual. Manag.</i> 26 (2015) 1188–1201. doi:10.1080/14783363.2014.920179.
S40	W. Thomas, A. Delis, V. Basili, An analysis of errors in a reuse-oriented development environment, <i>J. Syst. Softw.</i> 38 (1997) 211–224. doi:10.1016/S0164-1212(96)00152-5.
S41	P. Mohagheghi, R. Conradi, O.M. Killi, H. Schwarz, An Empirical Study of Software Reuse vs. Defect-Density and Stability, (2004) 282–292.
S42	G. Succi, L. Benedicenti, T. Vernazza, Analysis of the effects of software reuse on customer satisfaction in an RPG environment, <i>IEEE Trans. Softw. Eng.</i> 27 (2001) 473–479. doi:978 0 7340 3893 7.
S43	M. Morisio, D. Romano, I. Stamelos, Quality , Productivity , and Learning in Framework-Based Development : an exploratory case study, <i>IEEE Trans. Softw. Eng.</i> 28 (2002) 876–888.
S44	S. Isoda, Experiences of a software reuse project, <i>J. Syst. Softw.</i> 30 (1995) 171–186. doi:10.1016/0164-1212(94)00132-7.
S45	A. Ferrari, A. Fantechi, S. Gnesi, Lessons learnt from the adoption of formal model-based development, <i>Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)</i> . 7226 LNCS (2012) 24–38. doi:10.1007/978-3-642-28891-3_5.
S46	F. Mao, X. Cai, B. Shen, B. Jin, Operational Pattern Based Code Generation For Management Information System : An Industrial Case Study, (2016).
S47	D. Shirtz, M. Kazakov, Y. Shaham-Gafni, Adopting Model Driven Development in a Large Financial Organization, <i>Model Driven Archit. Found. Appl.</i> (2007) 172–183. doi:10.1007/978-3-540-72901-3_13.
S48	P. Baker, S. Loh, F. Weil, Model-Driven Engineering in a large industrial context—Motorola case study, <i>Model Driven Eng. Lang. Syst.</i> (2005) 476–491. doi:10.1007/11557432_36.
S49	T. Weigert, Practical Experiences in Using Model-Driven Engineering to Develop Trustworthy Computing Systems., <i>Sutic.</i> (2006) 208–217.

## 2. Borderline (excluded) studies

Although they were initially selected, after a full text reading the following seventeen studies were not considered for the inclusion in the set of selected primary works.

BP1	M. C. Franky, J. A. Pavlich-Mariscal, M. C. Acero, A. Zambrano, J. C. Olarte, J. Camargo, and N. Pinzón, "ISML-MDE," <i>Int. J. Web Inf. Syst.</i> , vol. 12, no. 4, pp. 533–556, Nov. 2016.
BP2	V. Bauer and A. Vetro, "Comparing reuse practices in two large software-producing companies," <i>J. Syst. Softw.</i> , vol. 117, pp. 545–582, 2016.
BP3	D. C. Rine and N. Nada, "Empirical study of a software reuse reference model," <i>Inf. Softw. Technol.</i> , vol. 42, no. 1, pp. 47–65, 2000.
BP4	B. Fitzgerald and T. Kenny, "Developing an information systems infrastructure with open source software," <i>IEEE Softw.</i> , vol. 21, no. 1, pp. 50–55, 2004.
BP5	J. S. Norris, "Mission-critical development with open source software: lessons learned," <i>IEEE Softw.</i> , vol. 21, no. 1, pp. 42–49, 2004.
BP6	M. F. Dunn and J. C. Knight, "Software reuse in an industrial setting: A case study," <i>Science (80- )</i> , pp. 329–338, 1991.
BP7	M. Irshad, R. Torkar, K. Petersen, and W. Afzal, "Capturing cost avoidance through reuse," <i>Proc. 20th Int. Conf. Eval. Assess. Softw. Eng. - EASE '16</i> , pp. 1–12, 2016.
BP8	M. Gasparic, A. Janes, A. Sillitti, and G. Succi, "An analysis of a project reuse approach in an industrial setting," <i>Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)</i> , vol. 8919, no. June 2012, pp. 164–171, 2014.
BP9	J. Varnell-Sarjeant, A. A. Andrews, and A. Stefik, "Comparing Reuse Strategies: An Empirical Evaluation of Developer Views," <i>2014 IEEE 38th Int. Comput. Softw. Appl. Conf. Work.</i> , pp. 498–503, 2014.
BP10	S. Mazzini, J. Favaro, and T. Vardanega, "Cross-Domain Reuse : Lessons Learned in a Multi-project Trajectory," pp. 113–126, 2013.
BP11	K. Sherif and A. Vinze, "Domain engineering for developing software repositories: a case study," <i>Decis. Support Syst.</i> , vol. 33, no. 1, pp. 55–69, 2002.
BP12	A. J. Nolan and S. Abrahão, "Dealing with cost estimation in software product lines: Experiences and future directions," <i>14th Int. Conf. Softw. Prod. Lines Going Beyond, SPLC 2010</i> , vol. 6287 LNCS, pp. 121–135, 2010.
BP13	M. Kircher and P. Hofman, "Combining systematic reuse with Agile development," <i>Proc. 16th Int. Softw. Prod. Line Conf. - SPLC '12 - volume 1</i> , vol. 1, p. 215, 2012.
BP14	N. Devos, C. Ponsard, J.-C. Deprez, R. Bauvin, B. Moriau, and G. Anckaerts, "Efficient reuse of domain-specific test knowledge: An industrial case in the smart card domain BT - 34th International Conference on Software Engineering, ICSE 2012, June 2, 2012 - June 9, 2012," <i>Proc. 34th Int. Conf. Softw. Eng.</i> , pp. 1123–1132, 2012.
BP15	H. Koziolok, T. Goldschmidt, T. de Gooijer, D. Domis, S. Sehestedt, T. Gamer, and M. Aleksy, "Assessing software product line potential: an exploratory industrial case study," <i>Empir. Softw. Eng.</i> , vol. 21, no. 2, pp. 411–448, 2016.
BP16	J. S. Cuadrado, J. L. Cánovas Izquierdo, and J. G. Molina, "Applying model-driven engineering in small software enterprises," <i>Sci. Comput. Program.</i> , vol. 89, no. PART B, pp. 176–198, 2014.
BP17	V. Bauer, J. Eckhardt, B. Hauptmann, and M. Klimek, "An exploratory study on reuse at google," <i>Proc. 1st Int. Work. Softw. Eng. Res. Ind. Pract. - SER&amp;IPs 2014</i> , pp. 14–23, 2014.

### 3. Data extraction form

This appendix shows the Microsoft Excel template we developed to assist the team in the data extraction process.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1		Research Space							Publication Space				
2	Paper_ID	RQ1	RQ2	RQ3	RQ4	RQ5	RQ6		PQ1	PQ2	PQ3	PQ4	PQ5
3	P1												
4	P2												
5	P3												
6	P4												
7	P5												
8	P6												
9	P7												
10	...												
11	Open List. Benefits reported (Verbatim)												
12													
13													
14	Open List. Reuse processes reported (Verbatim)												
15													
16													
17													

Figure 10 DEF for Research Space Questions

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1		Research Space							Publication Space					
2	Paper_ID	RQ1	RQ2	RQ3	RQ4	RQ5	RQ6		PQ1	PQ2	PQ3	PQ4	PQ5	
3	P1													
4	P2													
5	P3													
6	P4													
7	P5													
8	P6													
9	P7													
10	...													
11														
12														

Figure 11 DEF for Publication Space Questions

	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1		Research Space							Publication Space						
2	New_ID	RQ1	RQ2	RQ3	RQ4	RQ5	RQ6		PQ1	PQ1a	PQ2	PQ3	PQ4	PQ4a	PQ5
49	S2											Tomer, A.	I	RAFAEL Ltd., Haifa	Israel
50	S20	Quality	Not r	Hardware/Software	Systematic	Experience paper	No		IEEE Software	J	1995	Griss, M.L.	I	Hewlett-Packard Labs, San Jose, CA	USA
51	S20	Development time										Wosser, M.	I	Hewlett-Packard Labs, San Jose, CA	USA
52	S21	Quality	Reus	Software	Systematic	Evaluation research (Survey)	No		Proceedings of the Asia-Pacifi	C	1997	Butler, G.	A	Concordia University, Montreal	Canada
53	S21	Development cost													
54	S21	Productivity													
55	S22	Higher productivit	MBD	Software	Not reported	Evaluation research (Survey)	Yes		Journal of Systems and Softw	J	2013	Torchiano, M.	A	Politecnico di Torino	Italy
56	S22	Standardization										Tomassetti, F.	A	Politecnico di Torino	Italy
57	S22	Portability										Reggio, G.	A	Università di Genova	Italy
58	S22	Platform independence										Ricca, F.	A	Università di Genova	Italy
59	S22											Tiso, A.	A	Università di Genova	Italy
60	S23	Reduce Time-to-M Proje	Aerospace		Systematic	Evaluation research (Case Stud	Yes		Requirements Engineering	J	2015	Goldin, L.	A	Academic College of Engineering, Tel	Israel
61	S23	Development time										Berry, D.M.	A	University of Waterloo, Waterloo, O	Canada
62	S23	Productivity													
63	S24	Reduce Costs	with-	Not reported	Not reported	Evaluation research (Survey)	No		International Conference on S	C	2009	Land, R.	A	Mälardalen University, Västeras	Sweden
64	S24	Development time for-reuse										Sundmark, D.	A	Mälardalen University, Västeras	Sweden
65	S24											Causevic, A.	A	Mälardalen University, Västeras	Sweden
66	S24											Krasteva, I.	A	Sofia University, Sofia	Bulgaria
67	S24											Lüders, F.	A	Mälardalen University, Västeras	Sweden
68	S25	Time to market	CBD	Microelectronic	Systematic	Experience paper	No		Proceedings - 2012 Internatio	C	2012	Srivastava, R.	I	Freescale Semiconductor India Pvt L	India
69	S25	Productivity										Gupta, G.	I	Freescale Semiconductor India Pvt L	India
70	S25	Quality										Mudgil, N.	I	Freescale Semiconductor India Pvt L	India
71	S25	Reduce verification time										Mondal, H.	I	Freescale Semiconductor India Pvt L	India

Figure 12 An example of DEF with partial results

#### 4. Mapping process evaluation

This section contains an evaluation of the work done to “systematically develop” this SMS. It can be used as a self-evaluation, to help authors to check if everything has been done in the right way. The next table summarizes all the possible activities to consider when conducting a systematic mapping study.

##### Identified activities for conducting a Systematic Mapping Study. Adapted from[13]

Phase	Actions	Applied
Need for map	Motivate the need and relevance	✓
	Define objectives and questions	✓
	Consult with target audience to define questions	---
Study Identification		
Choosing search strategy	Automatic search (databases)	✓
	Snowballing	✓
	Manual (Conferences, Main Authors)	✓
Develop the search	PICO	✓
	Consult librarians or experts	✓
	Iteratively try finding more relevant papers	✓
	Keywords from known papers	✓
	Use standards, encyclopedias, and thesaurus	---
Evaluate the search	Test-set of known papers	✓
	Expert evaluates result	✓
	Search web-pages of key authors	✓
Inclusion/Exclusion	Identify objective criteria for decision	✓
	Add additional reviewer, resolve disagreements	✓
	Decision rules	✓
Data extraction and	Identify objective criteria for decision	✓
Classification	Obscuring information that could bias	---
	Add additional reviewer, resolve disagreements	✓
	Test–retest	---
	Classification scheme	✓
	Research type	✓
	Research method	✓

	Venue type	√
Validity discussion	Validity discussion/limitations provided	√

We applied the evaluation rubric suggested by Petersen [13] to evaluate our work in terms of all the key activities a SMS should include. The following tables show the rubric criteria. The scores identified by our mapping study are highlighted (bold text):

**Rubric: need for review.**

Evaluation	Description	Score
No description	The study is not motivated and the goal is not stated	0
Partial evaluation	Motivations and questions are provided	<b>1</b>
Full evaluation	Motivations and questions are provided, and have been defined in correspondence with target audience	2

**Rubric: choosing the search strategy.**

Evaluation	Description	Score
No description	Only one type of search has been conducted	0
Minimal evaluation	Two search strategies have been used	1
Full evaluation	Three or more search strategies have been used	<b>2</b>

**Rubric: evaluation of the search.**

Evaluation	Description	Score
No description	No actions have been reported to improve the reliability of the search and inclusion/exclusion	0
Minimal evaluation	At least one action has been taken to improve the reliability of the search OR the reliability of the inclusion/exclusion	1
Partial evaluation	At least one action has been taken to improve the reliability of the search AND the inclusion/exclusion	2
Full evaluation	All actions identified have been taken	<b>3</b>

**Rubric: extraction and classification.**

Evaluation	Description	Score
No description	No actions have been reported to improve on the extraction process or enable comparability between studies through the use of existing classifications	0
Minimal evaluation	At least one action has been taken to increase the reliability of the extraction process	1
Partial evaluation	At least one action has been taken to increase the reliability of the extraction process, and research type and method have been classified.	<b>2</b>
Full evaluation	All actions identified have been taken	3

**Rubric: study validity.**

Evaluation	Description	Score
No description	No threats or limitations are described	0
Full evaluation	Threats and limitations are described	<b>1</b>

Our systematic mapping obtained a final score: 9/11

We also calculated the ratio of the number of actions taken with respect to the total number. For this mapping study the ratio was  $22/26 = 84.6\%$